اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject: 1

Year. Month. Date. ( )

تحلیل زمانی:

```
Merge-Sort (A, p, r)                                   T(n)
if p<r
    q ← ⌊(p+r)/2⌋                                      c
    merge-Sort (A, p, q)                               T(n/2)
    merge_Sort (A, q+1, r)                             T(n/2)
    merge (A, p, q, r)                                 c1 n
```

تابع بازگشتی پس زمان اجرا هم بازگشتی

$T(n) = c_1 n + c + 2T(n/2)$

combine: $c_1(n)$ — $\theta(n)$ — $D(n)$

هزینه اجرای Merge-Sort

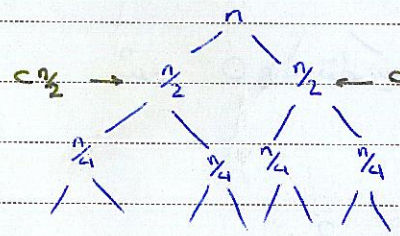
تابع خطی بر حسب n

Divide conquer الگوریتم

$T(n) = aT(n/b) + D(n) + C(n)$ ⇒ فرمول کلی

هزینه ترکیب — هزینه تقسیم

هر الگوریتم بازگشتی که به حالت پایه برسد و هزینه اضافات تمام می‌شود.



$c_n = c_{n/2} + c_{n/2}$

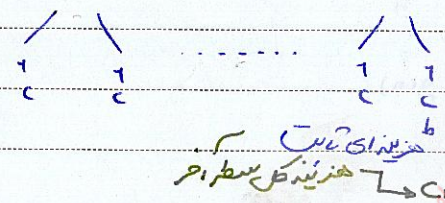
$4 \cdot \frac{cn}{4} = cn$

برای تقسیم هزینه ثابت c داریم

$c_1 n + c_2 \leq cn$

$\exists c, n_0 \; \forall n \geq n_0$

هزینه ترکیب و تقسیم: constant بر حسب n هست.



هزینه‌ای ثابت c

هزینه کل سطح i ام $\rightarrow c_n$

تعداد زیر مسائل سطح i ام: 2^i

هزینه هر زیر مسئله: $c \frac{n}{2^i}$

زمان کل هزینه تقسیم ترکیب برحسب n

وقتی یک عضو رسیدیم به از شرط پایه استفاده کنیم

call های بازگشتی را وسط داریم به پس تمام حالت پایه رسیدیم هزینه ترکیب و تقسیم را هم متقابلا حساب کردیم

جمع کل هزینه کل می‌شود

هزینه اجرای کل = هزینه برگ‌ها + مجموع هزینه سطوح

$= cn + \sum_{i=0}^{(\text{ارتفاع})-1} cn$

برگ‌ها را جداگانه حساب کنیم

برگ‌ها را جمع کنیم

$\frac{n}{2^i} = 1 \Rightarrow i = \log_2^n$

تعداد برگ‌ها $= 2^i = 2^{\lg n} = n$

$\rightarrow cn + cn(\log n - 1)$

جمله بزرگتر دارای رشد بیشتر

PAPCO
دانشکده فنی و کامپیوتر و فناوری اطلاعات دانشگاه علمی کاربردی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد و آل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

merge-sort برای الگوریتم $T(n) = \Theta(n \log n)$

تمرین 2.3-7) الگوریتمی که زمان اجرایش $n \log n$ باشد برای اینکه:

S | { عدد صحیح / عدد صحیح x

دو عضوی از S هست که مجموعشان x شود؟

$\exists a,b \in S \Rightarrow a+b=x$

پیشرفته:

صعبالعبوری توابع خود تابع هم باشد ⇒ برای صعبالعبوری توابع ($<$) ⇒ O بزرگ

" صعبالعبوری توابع ($>$) ⇒ Ω

برای همین ضرایب مختلف هم از بالا هم از پایین محدودند ($=$) ⇒ Θ

Θ ↳ هم O و هم Ω

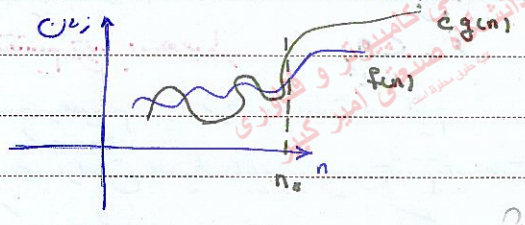
مثل big O فقط اینکه صعبالعبوری توابع ضعیف تابع باشد ($<$) ⇒ o کوچک

($>$) ⇒ ω

توابع مختلف هستند یا مثبت)

$f(n) = O(g(n)) \iff \exists c>0, n_0>0$

$\forall n \geq n_0 \quad f(n) \leq c\,g(n)$



مثال) $f(n) = 2n^2 + 10$ $\quad g(n) = n^2$ $\quad f(n) = O(n^2)$ $\quad c, n_0$

$\forall n \geq n_0 \quad 2n^2 + 10 \leq cn^2$

$g(n) = n^3$

$c = 4$, $n_0 = 3$ ⇒ $2n^2 - 10 \geq 0$

PqPCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject: 2

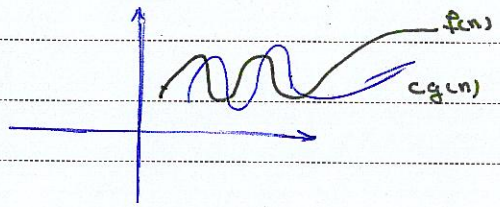
Year. Month. Date. ( )

$f(n) \neq O(n)$ ← هم n هم n^2 و $n\log n$ هم $n\log^n$ است.

$f(n) \neq O(n\log n)$

$f(n) = \Omega(g(n)) \iff \exists c > 0$

$n_0 > 0 \quad \forall n \geq n_0 \quad f(n) \geq c\,g(n)$



مثال) $f(n) = 2n^2 + 10$

$g(n) = n^2 \Rightarrow f(n) = \Omega(n^2)$

$g(n) = n\log n \Rightarrow f(n) = \Omega(n\log^n)$

$g(n) = n \Rightarrow f(n) = \Omega(n)$

قضیه:

$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n))$

از خط دوم باید بحرفی باشد یعنی دو تا از هم بیشتر باشد (f, g)

$f(n) = \Theta(g(n)) \iff \exists c_1, c_2 > 0, n_0 > 0$

$\forall n \geq n_0 \quad c_1 g(n) \leq f(n) \leq c_2 g(n)$

ضرایب باید constant بحرفی نداشته

مثال) $f(n) = 2n^2 + 10$

$g(n) = 3n^2 + 100 \Rightarrow 2n^2 + 10 = \Theta(3n^2 + 100)$

$g(n) = 100n^2 - 50 \qquad = \Theta(100n^2 - 50)$

$2n^2 + 10 \neq \Theta(n^3)$

$\neq \Theta(n)$

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

PAPCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

$f(n) = O(g(n)) \iff \forall c>0\ \exists n_0>0$

$\forall n \geq n_0 \quad f(n) < cg(n)$

$f(n)$ حد بالای $g(n)$ به ضریب ثابت c برای nهای بزرگتر از n_0 است. یعنی $f(n)$ از $cg(n)$ بیشتر نمی‌شود.

مثل $\leq$ می‌ماند Big O است برای یک حد

$f(n) = 2n^2 + 1$

$f(n) = O(n^2 \log n)$

$f(n) = O(n^3)$

$f(n) \neq O(n^2)$

$f(n) = \omega(g(n)) \iff \forall c>0 \quad \exists n_0>0$

$\forall n \geq n_0 \quad cg(n) < f(n)$

$f(n) = 2n^2 + 1$

$f(n) = \omega(n \log n) \qquad f(n) \neq \omega(n^2)$

$= \omega(n)$

$\rightarrow f(n) = O(g(n))$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

$f(n) = \omega(g(n))$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

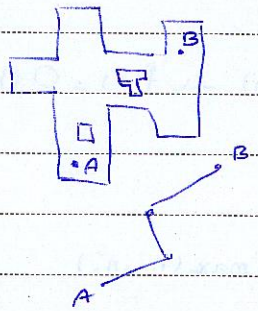
اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

هندسه‌ی محاسباتی computational Geometry

۱- برای حل مسایل در فضای دو بعدی الگوریتم بدهد

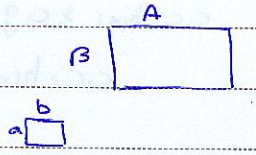


برای حرکت یک robot مسیری پیدا کند به موانع برخورد نکند

نیز جهت برای ربات مشکل است تغییر مسیر داشته باشد

mawen.smith.edu/~orourke/TOPP

pallet loading



مسائلی جهت بسته‌بندی کارتن برای توانیم به صورت انتخابی یا عمودی در آن (مستطیل بزرگ) قرار دهیم.

۱۰۰ نقطه در صفحه چند ضلعی ساده‌ی این

نقاط را بتوان ساخت.

lower bound و upperbound را پیدا کنیم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

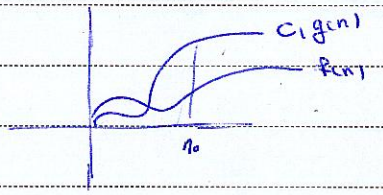
Subject:
Year. Month. Date. ( )

خاصیت ترانزیتی:

$f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$

$\Rightarrow f(n) = \Theta(h(n))$

$f(n) = O(g(n))$ and $g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$



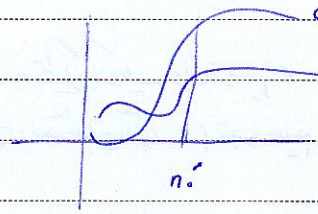
$n_0 = \max(n_0, n_0')$

$c' = c_1 c_2$

$c' h(n) \geq f(n)$

$c_1 c_2 h(n) \geq c_1 g(n) \geq h(n)$

$c_1 c_2 h(n) \geq c_1 g(x)$



برای هر 5 تا notation خاصیت ترانزیتی برقرار است.

خاصیت بازتابی:

$f(n) = \Theta(f(n))$

$f(n) = O(f(n))$

$f(n) = \Omega(f(n))$

constant ها را باید در نظر بگیریم برقرار است

برای O، اگر چپ برقرار است چون تابع و بیشتر از f تعریف شده

دانشکده مهندسی کامپیوتر و فناوری
اطلاعیه دانشگاه صنعتی امیر کبیر

خاصیت تقارنی:

$f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$

اگر و از بالا و پایین f را محدود کند g هم از بالا و پایین محدود می‌کند

این نقطه برای Θ است، برای مثال برای O برقرار نیست

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری، ارشد، کنکور، پیام نور، غیرانتفاعی، آزاد

Subject:
Year. Month. Date. ( )

خصیت بدتناری:

$f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$

$f(n) = o(g(n)) \iff g(n) = \omega(f(n))$

$O \leq , \Omega \geq , o < , \omega > , \Theta =$

$a, b \in \mathbb{R}$ ① $a > b$ ② $a < b$ ③ $a = b$

$f(n) \neq O(g(n))$

$f(n) \neq \Omega(g(n))$

$f(n) \neq \Theta g(n))$

$f(n) = n$, $g(n) = n^{1+\sin n}$

$\max(f(n), g(n)) = \Theta(f(n) + g(n))$

$h(x) = \begin{cases} f(x) & f(x) \geq g(x) \\ g(x) & g(x) \geq f(x) \end{cases}$

$(n+a)^b = \Theta(n^b)$

$f(x) + g(x) \geq 0 \Rightarrow f(x), g(x) \geq 0$

$f(x) + g(x) \geq g(x) \geq 0$

$f(x) + g(x) \geq h(x) \Rightarrow \max\{f(n), g(n)\}$

$f(x) < h(x)$

$g(x) < h(x)$

$f(x) + g(x) \leq 2h(x) \Rightarrow h(x) \geq \frac{1}{2}(f(x) + g(x))$ $c_1 = \frac{1}{2}$ c_2

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد .
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

یادآوری روابط ریاضی

$x \in \mathbb{R}$

$$x-1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x+1$$

$n \in \mathbb{N}$

$$\lfloor n/2 \rfloor + \lceil n/2 \rceil = n$$

$n \in \mathbb{R}$ $a,b>0$, $a,b \in \mathbb{N}$

$$\left\lceil \frac{\lceil n/a \rceil}{b} \right\rceil = \left\lceil \frac{n}{ab} \right\rceil$$

$$\left\lfloor \frac{\lfloor n/a \rfloor}{b} \right\rfloor = \left\lfloor \frac{n}{ab} \right\rfloor$$

$$\left\lceil \frac{a}{b} \right\rceil \leq (a+(b-1))/b$$

$$\left\lfloor \frac{a}{b} \right\rfloor \geq (a-(b-1))/b$$

$\lim_{n\to\infty} \frac{n^b}{a^n} = 0 \Rightarrow n^b = o(a^n)$ توابع $a,b \in \mathbb{R}$

رشد تابع‌های نمایی از تابع چندجمله‌ای رشد بیشتر است

$\log^b n$ توان

$\lim_{n\to\infty} \frac{\log^b n}{n^a} = 0 \Rightarrow (\log^b n) = o(n^a)$

$\log^n$ $n^{0.000\ldots001}$

رشد اولی بیشتر است

$$f^{(i)}(n) = \begin{cases} n & \text{if } i=0 \\ f(f^{(i-1)}(n)) & \text{if } i>0 \end{cases}$$

تابع f را با خودش ترکیب کنیم

$$\log^* n = \min\{ i \geq 0 : \log^{(i)} n \leq 1 \}$$

جذر، لگاریتم، $\log^*$ کندتر رشد می‌کنند

$\log^* 2 = 1$ $\log^* 4 = 2$ $\log^* 2^{(65536)} = 5$

$65536 = 2^{16} < 2^{80}$ تعداد اتم‌های دنیا

کمتر یا مساوی سریع خواهیم داشت که $\log^*$ آنها 5 باشد.

دانشکده مهندسی کامپیوتر
اطلاعات دانشگاه صنعتی امیرکبیر

PqPCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

اعداد فیبوناچی

$F_0 = 0$

$F_1 = 1$

$F_i = F_{i-1} + F_{i-2}$ if $i \geq 2$

↳ recursive

نسبت طلایی $\phi = \frac{1+\sqrt{5}}{2} \simeq 1.6$

$\hat{\phi} = \frac{1-\sqrt{5}}{2} \simeq -0.6$

$i \to \infty$ این جمله تقریباً برابر صفر می‌شود

غیر recursive $F_i = \frac{\phi^i - \hat{\phi}^i}{\sqrt{5}}$

فیبوناچی به صورت نمایی رشد می‌کند

تمرین 3.2.3

$n! = o(n^n)$ $\lim_{n\to\infty} \frac{n!}{n^n} = 0$

$n! = \omega(2^n)$ $\lim_{n\to\infty} \frac{n!}{2^n} = \infty$

تمرین 3.2

A	B	o	ω	Ω	Θ
$\lg^k n$	n^ε			X	X
n^k	c^n			X	X
$\sqrt{n}$	$n^{\sin n}$				
2^n	$2^{n/2}$	X		X	X
$n^{\lg c}$	$c^{\lg n}$			X	X
$\lg(n!)$	$\lg(n^n)$	X		X	X

$\varepsilon > 0$ و $k \geq 1$ و $c > 0$

تمرین 1

رایه از n عدد n! ترتیب‌های مختلف ، $\Theta(n!)$

دانشکده مهندسی کامپیوتر و فناوری
دانشگاه صنعتی امیر کبیر

```
if n==1
    return S
else
    S = perm(S, n-1)
    for (int i=0; i<n; i++)
        for(int j=i; j<n; j++
```

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

روابط بازگشتی:

حل ۳ روش برای روابط بازگشتی:
۱- روش جایگذاری: حدس برای حل داریم این حدس را با استفاده از استقرا ثابت می‌کنیم
۲- روش درخت بازگشت: این روش بیشتر برای حدس زدن جواب است (زیرمسئله‌ها را بسط می‌دهیم) هزینه تقسیم و ترکیب + هزینه برگ‌ها
۳- استفاده از قضیه اصلی: (قضیه‌ای که مستقیماً جواب اصلی را بدست می‌دهد)

زیرمسئله‌ها باید بسیار متفاوت باشند ← $T(n) = aT(n/b) + f(n)$ قضیه اصلی

روش جایگذاری:

حدس در روش جایگذاری را از ۲ طریق بدست می‌آوریم یکم بیشتر حدس می‌زنیم

مثال: $T(n) = 2T(\lfloor n/2 \rfloor) + n$ ⇒ merge sort $T(1) = 1$

حدس: $T(n) = O(n \log n)$

حکم: $T(n) \leq cn\log n$ (حکم)

فرض: $T(\lfloor n/2 \rfloor) \leq c\lfloor n/2 \rfloor \log \lfloor n/2 \rfloor$ فرض می‌کنیم این رابطه برای زیرمسئله‌ها برقرار است برای همین ثابت c را داریم

$$T(n) = 2T(\lfloor n/2 \rfloor) + n \leq 2c\lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor + n$$
$$\leq 2c\, n/2 \lg n/2 + n$$
$$\leq cn(\lg n - \lg 2) + n$$
$$\leq cn\lg n - cn + n \leq cn\lg n$$

$-cn + n \leq 0 \Rightarrow cn \geq n \Rightarrow c \geq 1$ (I)

$n = 2 \Rightarrow T(2) = 2T(\lfloor 2/2 \rfloor) + 2 = 2T(1) + 2 = 2 \times 1 + 2 = 4$

$T(2) = 4 \leq (c \times 2 \times \lg 2) = 2c \Rightarrow c \geq 2$ (II)

ترکیب (I) و (II) ⇒ $c \geq 2$ ⇒ هم پایه درست است هم استقرا

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر
P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

مثال) $T(n)=T(\lfloor n/2 \rfloor)+T(\lceil n/2 \rceil)+1$ $\quad T(1)=1$

حدس $T(n)=O(n)$

حکم $T(n)\le cn-b$ $\quad$ بیشتر حقی بجسب n

$O(n)$

برای n های مساوی درست است پس برای n های فرد هم درست است

فرض $\begin{cases} T(\lfloor n/2 \rfloor)\le c\lfloor n/2 \rfloor - b \\ T(\lceil n/2 \rceil)\le c\lceil n/2 \rceil - b \end{cases}$

$T(n)\le c\lfloor n/2 \rfloor + c\lceil n/2 \rceil + 1 - 2b$

$\le cn+1-b$ ($b\ge 1$) $\Rightarrow \le cn-b$ — این رابطه برقرار است — c هم نیست و هرچی باشد

حکم: $\le cn$ $\Rightarrow$ این برقرار نیست پس به حکمی رسیم چون c و n هر دو مثبت هستند

$-b+1\le 0 \quad \boxed{b\ge 1}$

۱- اگر به چنین جایی برسیم اگر یک const اضافه بود const حکم اضافه می‌کنیم

اگر به جمله اصلی درجه‌اش برابر بود نکتند، $\log^k$ اضافه می‌کنیم

اگر درجه‌اش از درجه حکم کمتر بود حکم درست است و متاسفانه اصلاح نشود

اگر درجه‌اش با حکم برابر بود باید یک $\log n$ به حکم اضافه کنیم

اگر بیشتر باشد حکم باید عوض شود و درجه را زیادتر کنیم

پایه $T(2)=T(\lfloor 3/2 \rfloor)+T(\lceil 3/2 \rceil)+1=3$

$T(2)\le 2c-b$

$3\le 2c-b \Rightarrow \frac{3+b}{2}\le c \Rightarrow b=1 \Rightarrow \boxed{c\ge 2}$

به راهی برای 1 درجه‌ی توان 2 درنظر گرفتیم

اگر درجه‌ی زمان نرمالتر حدس بزنیم درست است ولی بهترین حد پایین نیست.

← در مثال قبل اشکال اول اگر order را به اشتباه $O(n)$ حدس بزنیم داریم:

حدس $T(n)=O(n)$

حکم $T(n)\le cn$

فرض: $T(\lfloor n/2 \rfloor)\le c\lfloor n/2 \rfloor$

$T(n)\le 2c\lfloor n/2 \rfloor + n$

$\le 2cn/2 + n$

$\le cn+n$

$\le (c+1)n$

$\le cn$ $\quad$ X

درجه‌ی این حکم باید چه باشد؟ $\log$ بگیریم

می‌توانیم c را یک عدد بزرگ حساب کنیم درجه‌ی n مشکل ندارد

یک راه استراحت‌تر پیدا کنیم که c به هیچ‌جا وابسته نباشد

PAPCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

تقسیم و توان

مثال) $T(n) = 2T(\sqrt{n}) + \lg n$

$n = 2^m$ (تغییر متغیر) $\Rightarrow m = \lg n$

$T(2^m) = 2T(2^{m/2}) + m$

$S(2^m) = S(m)$ تغییر متغیر

$S(m) = 2S(m/2) + m$

$S(m) = O(m \lg m)$

$T(2^m) = O(m \lg m)$ باید تغییر متغیرها را برگردانیم

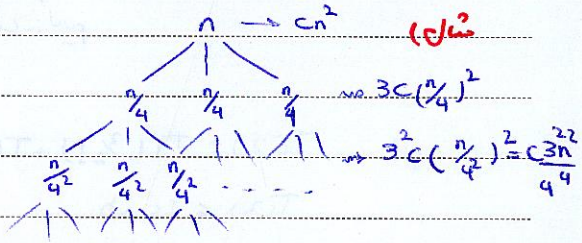
$T(n) = O(\lg n \lg\lg n)$

تمرین 4.1-6 $T(n) = 2T(\sqrt{n}) + 1$

روش درخت بازگشت:

اگر بخواهیم حساب کنیم جواب دقیق بازگشتی چیست (اثبات است) اگر حدس بزنیم چیست و بخواهیم با استقرا ثابت کنیم

مثال) $T(n) = 3T(n/4) + cn^2$ هزینه تقسیم و ترکیب



هزینه کل: $T(n) = \sum$ هزینه سطوح + هزینه برگ‌ها

تعداد گره‌های سطح i: 3^i

هزینه هر گره: $c \times (\frac{n}{4^i})^2 = \frac{cn^2}{16^i}$

هزینه سطح i: $3^i \times c(\frac{n^2}{16^i}) = cn^2(\frac{3}{16})^i$

سایز مسئله در سطح i: $\frac{n}{4^i} = 1 \Rightarrow i = \log_4^n$ سایز مسئله

تعداد سطوح برگ‌ها (ارتفاع درخت)

تعداد برگ‌ها: $3^i = 3^{\log_4^n} = n^{\log_4^3}$

هزینه کل: $T(n) = \sum_{i=0}^{\log_4^n - 1} cn^2(\frac{3}{16})^i + cn^{\log_4^3} = cn^2(\frac{1-(\frac{3}{16})^{\log_4^n}}{1-\frac{3}{16}}) + cn^{\log_4^3}$

$< \sum_{i=0}^{\infty} cn^2(\frac{3}{16})^i + cn^{\log_4^3} \leq \frac{cn^2}{1-\frac{3}{16}} + cn^{\log_4^3} < \frac{16}{13}cn^2 + cn^{\log_4^3}$ جمله غالب

$= O(n^2)$

PqPCO
تکثیر شده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

مثال (درخت بازگشت)

$T(n) = T(n/3) + T(2n/3) + O(n)$, $T(1) = c$

$T(n) \leq \sum_{i=0}^{\log_{3/2}^{n}-1} cn + n^{\log_{3/2}^{2}} \times c$

هزینه یک برگ × تعداد برگها

جمله اول برای T

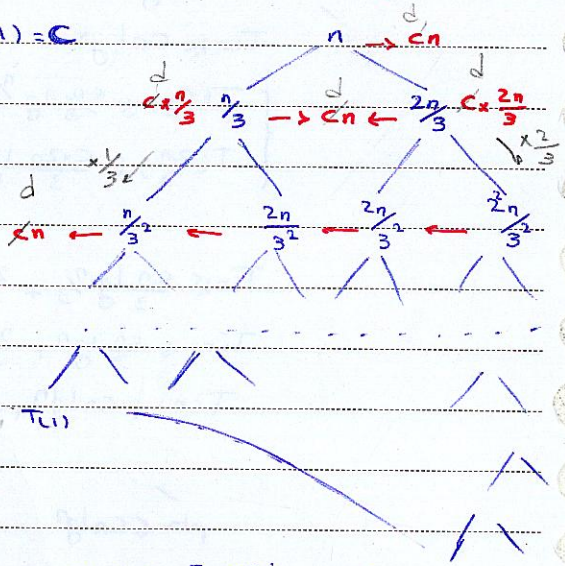
تعداد برگها $\leq 2^{\log_{3/2}^{n}} = n^{\log_{3/2}^{2}}$

رشد این جمله بیشتر است

$T(n) \leq cn \times \log_{3/2}^{n} + cn^{\log_{3/2}^{2}}$

$n\log_{3/2}^{n} \leq n^{\log_{3/2}^{2}}$ (اثبات)

$T(n) = O(n^{\log_{3/2}^{2}})$



سمت چپ درخت زودتر به 1 می‌رسد

$\frac{n}{3^i} = 1 \Rightarrow i = \log_3^n$ ← سمت چپ

$\frac{2^i n}{3^i} = 1 \Rightarrow i = \log_{3/2}^{n}$ ← " راست

ارتفاع سمت راست (عمق)

پس ما می‌خواهیم حدس را بیدا کنیم که ست ضعیف‌تر از برابر شاخه بزرگتری باشیم

این " باشد " " " " " " " " کوچکتر "

دو تا تابع log از نظر رشد با هم یکی هستند.

$\log_a^n = \log_b^n * \log_a^b$

کم رشد ترین حدس = بهترین حدس

(از کدام جایی بگیرم)

این بهترین حدس نیست (tight نیست) درخت را بزرگ کردیم تا آخر هزینه‌ها cn نیست

تعداد برگها از این کمتر است تا بگی که بهترین حدس وجود دارد حدس بالاتری آوردیم $n\log^n$

می‌شود کندترین رشدان $n\log^n$ است

بهترین حدس $n\log^n$ در نظر گرفتیم الرشان دهیم به این نتیجه رسیدیم این حدس tight است

پایین‌ترین که می‌زنیم احتمالاً حدسی که درست می‌آوردیم tight نیست

دانشکده مهندسی کامپیوتر و فناوری
اطلاعیه دانشگاهی صنعتی امیر کبیر

P4PCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

با استقرا ثابت می‌کنیم $T(n) = O(n \lg^n)$

حکم $T(n) \leq cn\lg n$

$$\begin{cases} T(n/3) \leq \frac{cn}{3}\lg\frac{n}{3} \\ T(\frac{2n}{3}) \leq \frac{c\times 2n}{3}\lg\frac{2n}{3} \end{cases}$$

$$T(n) \leq \frac{cn}{3}\lg\frac{n}{3} + \frac{2cn}{3}\lg\frac{2n}{3} + dn$$

$$T(n) \leq \frac{cn}{3}\lg n + \frac{2cn}{3}(\lg^2 + \lg n) + dn - \frac{cn}{3}\lg 3 - \frac{2cn}{3}\lg 3$$

$$T(n) \leq cn\lg n + \underbrace{dn + \frac{2cn}{3} - cn\lg 3}_{\leq 0}$$

حکم $\leq cn\lg n$

$$d + \frac{2}{3}c + \lg 3 \times c \leq 0 \Rightarrow c > \frac{d}{\lg 3 - \frac{2}{3}}$$

روش استفاده از قضیه اصلی Master Theorem

قضیه رو فقط برای توابعی که به این شکل باشند می‌تونیم بگیم

$T(n) = aT(n/b) + f(n)$

$b>1, a\geq 1$

بعد در این حالت‌ها a و b رو در جمله n/b جایگذاری می‌کنیم

از نظر مجانبی $f(n)$ با $n^{\log_b^a}$ چه رابطه‌ای دارد؟

$f(n) = O(n^{\log_b^a})$ برقرار است؟ (یعنی $f(n)$ از $n^{\log_b^a}$ مجانبی کوچکتر است)

حالت اول: اگر $f(n) = O(n^{\log_b^a - \varepsilon})$, $\varepsilon > 0$ بسیار کوچک $\Rightarrow T(n) = \Theta(n^{\log_b^a})$

حالت دوم: اگر $f(n) = \Theta(n^{\log_b^a}) \Rightarrow T(n) = \Theta(n^{\log_b^a} \times \lg^n)$

حالت سوم: اگر $f(n) = \Omega(n^{\log_b^a + \varepsilon})$, $\varepsilon > 0$, $af(n/b) \leq cf(n)$, $c<1$ $\Rightarrow$ $T(n) = \Theta(f(n))$

تمرین 9.2-4

PAPCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

مثال ۱)

$T(n) = 9T(n/3) + n$

$a=9 , b=3 , n^{\log_b a} = n^{\log_3 9} = n^2$

$f(n) = n$

$f(n)$ را با n^2 مقایسه می‌کنیم

$f(n) = n \stackrel{?}{=} O(n^{2-\varepsilon})$ ← به جای ε هر عددی را می‌توانیم بدهیم

$2-\varepsilon \geq 1 \Rightarrow \boxed{\varepsilon \leq 1}$

طبق قضیه اصلی حالت اول $\Rightarrow T(n) = \theta(n^{\log_b a}) = \theta(n^2)$

مثال ۲)

$T(n) = T(\frac{2n}{3}) + 1$

$f(n) = 1 \quad ? \quad n^{\log_{3/2} 1} = n^0 = 1$ ← هم مقایسه می‌کنیم

$f(n) = 1 = \theta(n^{\log_b a}) = \theta(n^{\log_{3/2} 1}) = \theta(1)$

طبق حالت دوم قضیه اصلی $T(n) = \theta(n^{\log_b a} \lg n) = \theta(1 \times \lg n) = \theta(\lg n)$

مثال ۳)

$T(n) = 3T(n/4) + n\lg n$

$f(n) = n\lg n \quad ? \quad n^{\log_4 3}$

$\log$ توان ۱ + یک ← توان ۱ < ← توان سمت راست بیشتر است ← توان سمت چپ کمتر است

$f(n) = n\lg n \stackrel{?}{=} \Omega(n^{\log_4 3 + \varepsilon})$

ε را چه عددی می‌دهیم؟ ← $1+\varepsilon$ ← 1 برقرار است

$n\lg n$ از n بیشتر است پس برقرار است

$\log_4^3 + \varepsilon \leq 1 \Rightarrow \varepsilon \leq 1 - \log_4^3$

شرط منظم بودن هم چک کنیم

$af(n/b) \leq cf(n)$

$3\frac{n}{4}\lg n/4 \leq cn\lg n$

$\frac{3n}{4}\lg n - \frac{3n}{4}\lg 4 \leq cn\lg n$

$c \geq \frac{3}{4}$

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

طبق حالت سوم رابطه $T(n) = \Theta(f(n)) = \Theta(n \lg n)$

مثال)

$T(n) = 2T(n/2) + n \lg n$

$f(n) = n \lg n$

I

$n^{\log_b^a} = n^{\log_2^2} = n$

II

ستد I و II را با هم مقایسه می‌کنیم

$n \lg n$ بیشتر از مرتبه چند جمله‌ای است (اما کمتر از)

$f(n) = n \lg n \stackrel{?}{=} \Omega(n^{\log_b^a + \varepsilon}) = \Omega(n^{1+\varepsilon})$

n^{ε} ی نمی‌تواند جوری پیدا شود که $n \lg n$ از n بیشتر شود

در نتیجه هیچ ε جوری نداریم که داشته باشیم

$n \lg n = \Omega(n^{1+\varepsilon})$

بنابراین نمی‌توانیم از قضیه اصلی استفاده کنیم چون فرض برقرار نیست

تمرین 9.4-2

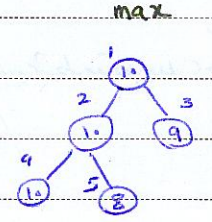
الگوریتم Heap Sort

از ساختمان داده‌ای بنام Heap برای Sort کردن استفاده می‌کند

Heap ← یک درخت تقریباً کامل دودویی که یک شرط خاص دارد

max Heap ← مقدار هر گره بزرگتر مساوی مقدار فرزندان

min Heap ← " " " کوچکتر " "



برای درست کردن Heap یک ساختمان داده‌ی دیگی بنام pointer

همچنین یک آرایه هم داریم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

از بالا به پایین از چپ به راست حرکت می‌کنیم

10	10	9	10	8

خانه‌ی فرزند چپ $2i$

فرزند راست $2i+1$

پدر $\lfloor i/2 \rfloor$

فرزند چپ:

LEFT (i) = 2i

RIGHT (i) = 2i+1

PARENT (i) = $\lfloor i/2 \rfloor$

طول آرایه را با Length[A] نشان می‌دهیم

← تعداد خانه‌های آرایه

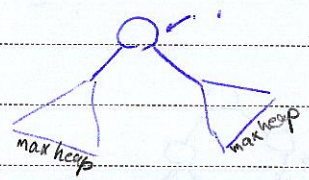
Heap_Size [A] ≤ تعداد خانه‌هایی که استفاده شده یعنی تعداد عددهای heap در آرایه

A[PARENT(i)] ≥ A[i] ← max heap

یعنی خانه‌ی پدر از چپ و راست بزرگتر می‌شود و برعکس وجود ندارد.

الگوریتم max-Heapify

یک درخت باینری که فرزند راست و چپش heap هست را می‌خواهیم که درخت max heap شود.



برای این که درخت فرزندانش این خاصیت را داشته باشد

الگوریتم از فرزندانش بزرگتر جستجو می‌کند و کاری می‌کنیم خودش max heap است

الگوریتم از فرزندانش بزرگتر جستجو می‌کند و کاری می‌کنیم خودش

دانشگاه صنعتی امیرکبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

```
max_Heapify (A, i)
l ← LEFT (i)
r ← RIGHT (i)
if (l ≤ heapsize[A] and A[l] > A[i])
then (largest ← l)
else (largest ← i)
if (r ≤ heap-size[A] and A[r] > A[largest])
then largest ← r
if largest ≠ i then
   exchange (A[i], A[largest])
   max_Heapify (A, largest)
```

max بین ریشه و فرزندانش را پیدا می‌کنیم

اگر فرزند راست بزرگتر بود

heap ← largest = i ست می‌کنیم

هزینه اجرا:

$T(n) \leq$ تعداد دفعات $\times \Theta(1)$ هزینه ثابت

$T(n) = O(\lg n)$

برای هزینه constant درخت زیرمجموعه به صدا زدن max_Heapify

$T(n) = T(\frac{2n}{3}) + c$

تعداد برگ‌های هر زیردرخت $\frac{n}{2}$

$3x = n-1$

$x = \frac{n-1}{3}$

$\frac{n}{2}$ و $\frac{2n}{3}$

$T(n) = c$? $n^{\log_b^a} = n^{\log_{3/2}^1} = n^0 = 1$

$c = \Theta(1) \Rightarrow T(n) = \Theta(1 \times \lg n)$

حالت دوم قضیه اصلی $= \Theta(\lg n)$

PqPCO

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

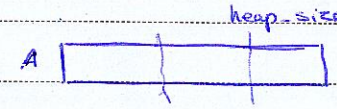
الگوریتم ساخت Heap :

با استفاده از max-Heapify

ترتیب رخ دادن توضیحاتی که حتماً Heap است برای max-Heapify را روی کاغذ آماده کنیم

و سعی کنیم با مثال توضیح دهیم

اگر گره برگ است $2i, 2i+1 > heap\_size[A]$




```
Build_max_Heap(A)
heap_size[A] ← length[A]
for i ← [length[A]/2] down to 1
    max_Heapify(A,i)
```

برای برگ‌ها کاری انجام نمی‌دهد

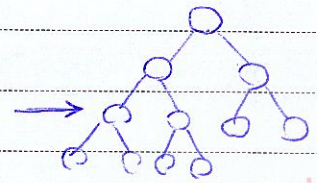
اگر $\lg^0$ است ← $\lg^0$

$T(n) = O(n \lg n)$

کران tight پیدا کنید.

max-Heapify(A,i) هزینه $\leq c \times$ ارتفاع i

هر چه از برگ‌ها بالا می‌رویم هزینه زیاد (تقاریبی) می‌شود.



تعداد گره	ارتفاع گره
1	$\lg n$
2	$\lg n - 1$
4	$\lg n - 2$
$n/8$	2
$n/4$	1

ناحیه آبی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

max-Heapify هزینه $T(n) = \sum_{i=1}^{\lg n} \left(\frac{n}{2 \times 2^i}\right) \times i^0 \times c$

PAPCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

$$T(n) = \frac{cn}{2} \sum_{i=1}^{\lg n} i\left(\frac{1}{2}\right)^i < \frac{cn}{2} \sum_{i=0}^{\infty} i\left(\frac{1}{2}\right)^i$$

$$\sum_{k=0}^{\infty} k x^k = \frac{x}{(1-x)^2} \qquad |x| < 1$$

$x = \frac{1}{2}$

$$T(n) < \frac{cn}{2} \times \left(\frac{\frac{1}{2}}{(1-\frac{1}{2})^2}\right) < \frac{cn}{2} \times 2$$

$$T(n) < cn \rightarrow T(n) = O(n)$$

For

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

PqPCO

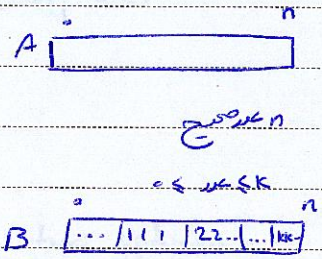
. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

Counting Sort

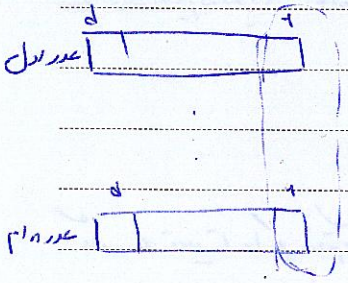


بدیهی داریم که جنس ۱ و جنس ۲ ... جنس k داریم

Radix Sort

n عدد صحیح d رقمی در مبنای k

اگر متوسط رقم‌ها مساوی d شود از این روش استفاده می‌کنیم

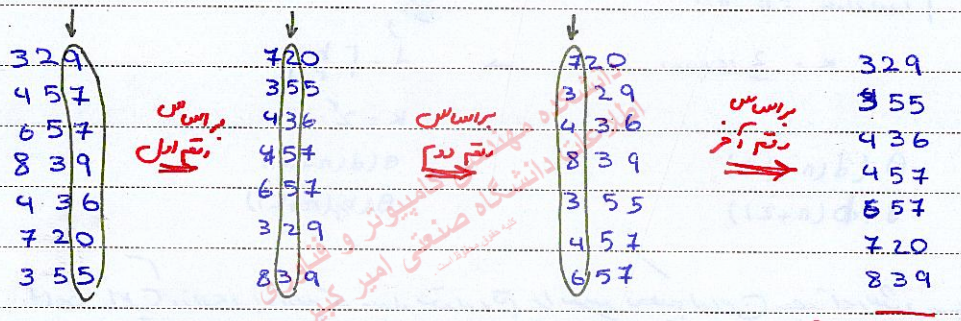


شکل اول مرتب براساس رقم اول مرتبط می‌باشد

... دوم ... دوم ...

و هم چنین رقمی که آخرین رقم مرتب شده کل را بدهد

مرتب شده است.



اگر در مرتب کردن دو رقم مساوی با هم ترتیب را بخواهیم رقمی که اول آمده در لیست جدید نیز اول می‌آید ← رقمی در عدد مساوی‌اند آن دو عدد را جابجا می‌کند

42 24 ⇒ 24 42

در حالت سوم رقم دوم اگر یکی بزرگتر بود باید رقم اول کدام بزرگتر است کاری نداریم ببینیم جابجا کنیم

P4PCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

تمرین) الگوریتم radix sort را تشریح کنید که اعداد n رقمی را از ارقام با ارزش تر به سمت رقم های کم ارزش انجام شود.

```
Radix_Sort (A,d)
for i ← 1 to d do
    use a stable sort to sort → counting sort  Θ(n+k)
    array A on digit A
```

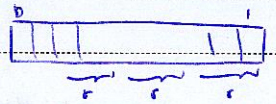
چون d بار این کار (sort) انجام میشود

$\rightarrow \Theta(d(n+k))$

اگر مبنای اعداد شامل n^2 شود ← پیچیدگی خطی نیست که $\Theta(n)$ باشد

← می‌توانیم در این sort رقم‌ها را با هم ترکیب کنیم و مبنای را عوض کنیم

2 در مبنای b_bit



که در آن یک رقم یک بیت می‌شود در مبنای 2^r

تعداد بیت = b_bit

$k = 2^r$ در مبنای ← $d = \lceil \frac{b}{r} \rceil$

$\Theta(d(n+k))$ یعنی $\Theta(\frac{b}{r}(n+2^r))$

$\Theta(b(n+2))$

در این sort اینجا داریم که مبنا و تعداد رقم را انتخاب کنیم این بهینه می‌شود ←

اگر مبنا را عوض کنیم شاید بتوانیم الگوریتم سریعتری برای آن پیدا کنیم

تمرین 8.3-4

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

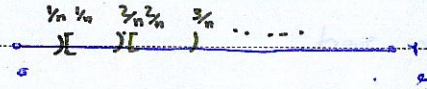
دقتی زیع معلوم است می‌توان از این استفاده کرد

Bucket Sort

برای اعداد حقیقی که در range [0,1) هستند از الگوریتم Bucket می‌توان

استفاده کنیم چون توزیع یکنواخت است میانگین زمان اجرا خطی است ← $\theta(n)$

اساس کار: بازه [0,1) را به n تا زیربازه تقسیم می‌کنیم



به دلیل یکنواختی انتظار داریم که در بازه‌های متناظر با اعداد تراکم زیاد

بنابراین زیربازه‌های این آرایه B خانه دارم

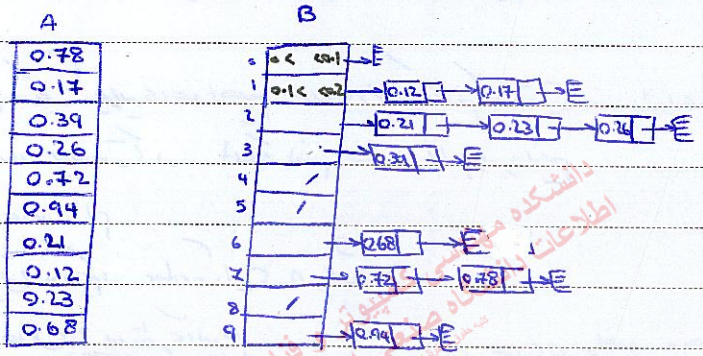
هر کدام از خانه‌ها یک لیست پیوندی است که ترتیبی دهیم

هر کدام از این لیست‌ها را با مثلاً insertion sort مرتب می‌کنیم ($\theta(n^2)$)

چون تعداد اعداد به علت یکنواختی برابر است ← $\theta(c)$

و چون n تا از این لیست داریم که باید مرتب کنیم ← $n \times \theta(1) = \theta(n)$

مثال ←



بازه‌ها هرچه‌ای تواند باشد مثلاً اگر بدانیم که بازه‌ی اعداد [0,100] است می‌توانیم زیربازه‌های با طول

1 داشته باشیم محدب است که بازه اعداد ثابت باشد

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

```
Bucket_Sort (A)
n ← length[A]
for i ← 1 to n do
    insert A[i] into list B[⌊n*A[i]⌋]

for i ← 0 to n-1 do
    sort list B[i] with insertion sort
concatenate the lists B[0], B[1], ..., B[n-1] together in order
```

bucket برای نگهداری اعداد صحیح بازه‌ای است.

اسلایدی ← $E(T(n)) = \Theta(n)$

تمرین 8.4-4

The Selection Problem

اگر متغیر داشته باشیم این کوچکترین = min

این ... = max

این کوچکترین را باید پیدا کنیم

متغیر داریم یک آرایه‌ای خواهیم عددی را پیدا کنیم که دقیقا از i-1 عنصر بزرگتر باشد

یک راه ساده این است که sort کنیم عنصر i ام را برداریم $\Omega(n \lg n)$

دنبال الگوریتمی بگردیم که order آن n باشد

حالت‌های خاص را بررسی کنیم

min ← i=1

max ← i=n

برای اینکه min و max را بیابیم حداکثر 2n-2 مقایسه نیاز داریم که بهتر هم می‌شود.

دسته اول را جفت مقایسه کنیم یک min و یک max داریم دسته دوم هم همین طور تا بعد برای پیدا کردن min بیا min ها را جفت مقایسه کنیم

$T(n) = 2(T(n/2) + \Theta(1))$

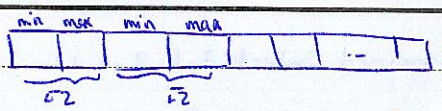
$T(n) = \Theta(n)$

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )



$\frac{n}{2}$

مقدار ماکسیمم و مینیمم را محاسبه می‌کنیم و حاصل دو تای حاصل را جمع تقسیم می‌کنیم

$\frac{n}{2} + (\frac{n}{2} - 1) \times 2 = \frac{3n}{2} - 2$

$\frac{n}{2} + \frac{n}{4} + (\frac{n}{4} - 1) \times 2$

$\frac{3n}{4} + \frac{2n}{4} - 2 \rightarrow \frac{5n}{4}$

تمرین 9.1-1

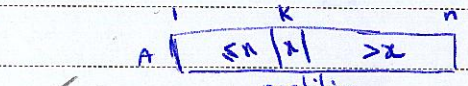
Selection Problem

بیان دقیق به طوری که دقیقاً i امین عدد از آن کوچکتر باشد

$\Omega(n \lg n)$

1- sort

2- انتخاب i امین



sort نیستند نسبت به هم ولی نسبت به x هستند ← sort نیست

اگر k شد سمت راست می‌توان حدس زد i است یا i عدد از x کوچکتر هستند

k=i است

Randomized_Select (A,p,r,i)

if p=r then → عنصر مورد نظر پیدا شده i=k

return A[p] → عنصری که نشستیم اگر i ما هم بود برگردانده شود

q ← Randomized_partition (A,p,r) → یعنی با random سمت راست می‌ریزیم

می‌رویم تقسیم‌بندی می‌کنیم عناصر کوچکتر آن در سمت راست و عناصر بزرگتر سمت چپ

k ← q-p+1

if i=k then return A[q]

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیرکبیر

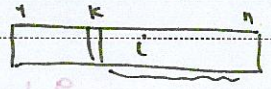
. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

else i<k then return randomized Select(A,p,q-1,i)
else return randomize_Select(A,q+1,r,i-k)

به عنوان مثال i-k امین عضو بزرگ در زیر آرایه ای است که از آن i-k است زیرا قبل از آن k تا عضو وجود داشته و وقتی به k+ شروع می شود i امین عضو در آرایه اصلی



شبه زبان اجرای quick sort وابسته به نوع partition است.

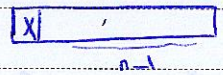
حالت

متوازن partition : → بهترین حالت

$T(n) = \theta(n) + T(n/2) = \theta(kn) = \theta(n)$ ← تست ۳ قضیه اصلی

در زیر آرایه بازگشتی متناظر با زیر آرایه ای داریم با چپ یا راست درصدی کمتر در quick sort هم چپ زیر آرایه ای شده هم راست

نامتوازن partition :



$T(n) = \theta(n) + T(n-1) = \theta(n^2)$

این زمانی پیش می آید که متوازن نباشد این partition هرکسری از سایز نباشد متوازن است و ضمنا زمان متوازن نیست که یک طرف constant نباشد این که هر کسری باشد ضمنا در پیاده سازی ریتم تغییری ندارد

سایز partition : pivot نسبت عدد pivot است در شخصی پیدا نکنم که سایز نباشد در این صورت تقریبا partition نامتوازن خواهد بود.

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

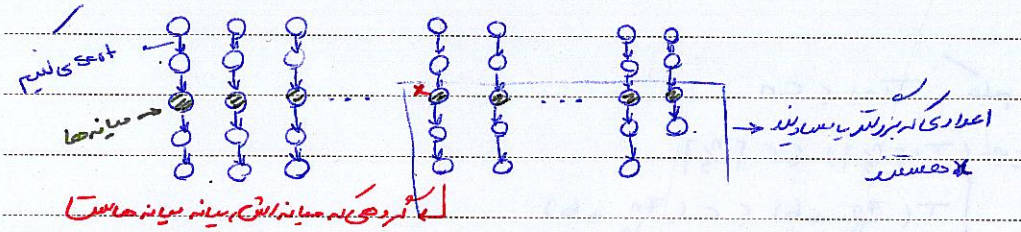
Subject:

Year. Month. Date. ( )

الگوریتمی که همیشه حالت بدترش رو تضمین می‌کند، order آن همیشه n است

الگوریتم Selection

1. اعداد را به $\lceil \frac{n}{5} \rceil$ گروه 5تایی (با اضافه احتمالاً یک گروه که کمتر از 5 است) تقسیم می‌کنیم
2. میانه هر یک از $\lceil \frac{n}{5} \rceil$ گروه را پیدا می‌کنیم (مثلاً با استفاده از insertion sort)
3. الگوریتم select را به صورت بازگشتی برای پیدا کردن میانه‌های فوق فراخوانی می‌کنیم
   حال کار select همین است
4. با یک دور با استفاده از جواب گام 3 (x = میانه میانه‌ها) partition می‌کنیم فرض کنید از سر آن k باشد.
5. اگر $k = i$ باشد x را برگردانیم. اگر $k > i$ الگوریتم select را بصورت بازگشتی برای نیمه اول فراخوانی می‌کنیم، اگر $k < i$ الگوریتم select را برای نیمه دوم، و برای پیدا کردن $k-i$ این عضو فراخوانی می‌کنیم



تمام گروه‌هایی که میانه‌اش از x بزرگتر است در سمت چپ و بقیه‌ای که بزرگتر و در سمت راست می‌آوریم

x میانه است ← نصف گروه‌ها بزرگتر از x

... کوچکتر از x

$\left(\frac{n/5}{2}\right) * 3 = \frac{3n}{10}$ ← حداقل

$\frac{3n}{10} - 6 \leq \quad \leq \frac{7n}{10} + 6$

آشنایی با مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

Select

زمان اجرای الگوریتم را تحلیل می‌کنیم

$T(n) = O(n) + T(\lceil n/5 \rceil) + T(\frac{7n}{10} + 6)$

۱ خط $O(n)$

۲ خط $O(n)$

۳ خط بازگشتی $T(\lceil n/5 \rceil)$

partition ۴ خط $O(n)$

۵ خط بازگشتی $T(\frac{7n}{10} + 6)$ (اندازه بزرگترین زیرمسئله که الگوریتم فراخوانی می‌شود)

← max وقت حداکثر را در نظر می‌گیریم

$\frac{3n}{10} - 6 \leq$ اندازه هر زیرمسئله $\leq \frac{7n}{10} + 6$

۶ خط بازگشتی ۵ تا از زیرمسئله‌ها کوچکتر هستند

این کارهایی که انجام دادیم تا الگوریتم balance شود و آن $O(n)$ شود

حکم $T(n) \leq cn$

فرض استقرا $\begin{cases} T(\lceil n/5 \rceil) \leq c\lceil n/5 \rceil \\ T(\frac{7n}{10} + 6) \leq c(\frac{7n}{10} + 6) \end{cases}$

$T(n) \leq an + c\lceil \frac{n}{5} \rceil + c(\frac{7n}{10} + 6)$ ($\lceil \frac{n}{5} \rceil \leq \frac{n}{5} + 1$)

$\leq an + c(\frac{n}{5} + 1) + c\frac{7n}{10} + 6c$

$\leq an + \frac{9cn}{10} + 7c$ ← برای اینکه حکم یعنی cn نتیجه دهیم برای سمت راست

$\leq cn + (\frac{-cn}{10} + 7c + an)$ ← $\frac{1}{10}cn$ را اضافه و کم می‌کنیم

برای ثابت کردن حکم باید این را ثابت کنیم ≤ 0

$\frac{-cn}{10} + 7c + an \leq 0$

$c(\frac{n}{10} - 7) \geq an$

دانشکده مهندسی کامپیوتر اطلاعات دانشگاه صنعتی امیرکبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

فرض $n \geq 70$

$c \geq 1.a \frac{n}{n-70}$

برای خواهیم یک c و یک n_0 پیدا کنیم که از n_0 به بعد همیشه برقرار باشد

برای تعیین n_0 این گونه عمل می‌کنیم

$n = 140 \Rightarrow c = 2.1.a$

$\forall n \geq 140 : \frac{n}{n-70} \leq 2$

در اینجا $n_0 = 140$ است که این رابطه برقرار است

$c \geq 1.a \times 2 \geq 1.a \frac{n}{n-70}$

برای هر a ثابت (constant) می‌توانیم یک c پیدا کنیم که این رابطه برقرار باشد

تمرین 9.3-8 ، 9.3-9

روش‌های حل مساله:

1- روش تقسیم و حل Divide & conquer

2- برنامه‌نویسی پویا dynamic programming

3- روش حریصانه Greedy

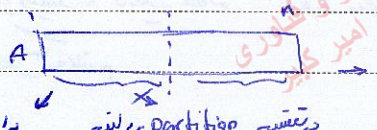
4- جستجوی فضای حالات: backtracking

گام‌های اصلی روش تقسیم و حل: (حالت کلی بازگشتی است)

1- تقسیم به زیرمسایل: مسئله‌ی خود را به چه صورتی تقسیم کنیم بهتر است

2- حل بازگشتی زیرمسایل: زمانی که مسایل حل شدند دیگر به بازگشت برنمی‌گردیم

3- ترکیب جواب زیرمسایل و برگشت از درون جواب اصلی



برای merge sort نصف می‌کنیم

در تقسیم یا partition می‌کنیم که حتماً نباید نصف باشد و یک عنصر به عنوان pivot داریم

برگشت در مسئله در راستای جواب اصلی

برای هر مسئله‌ای تشخیص دهیم که چطوری باید تقسیم کنیم

عمده کار در این روش گام 1 و 2 است.

P4PCO

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

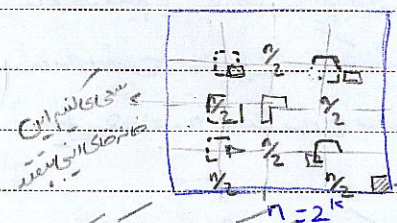
Subject:

Year. Month. Date. ( )

در مسایل بعد خصوصیت بصری مسئله کلی باشد

مثال: یک مربع داریم و فرض می‌کنیم $n = 2^k$

با حذف‌هایی برابر شکل ۱ به ۱ تبدیل کنیم



خانه‌های داخلی باقیند

خانه‌های این سطر و ستون حذف می‌شوند

برای حل مسئله تقسیم و حل: ابتدا خانه‌های حاشیه را حذف کنیم و خانه‌های باقی‌مانده را یک مسئله کلی

کاشی‌سازی

سپس کاشی‌سازی را تقسیم کنیم که قطعات هم‌نوع مسئله اصلی مربع شوند

$$\text{مساحت صفحه} = 2^k \times 2^k = 4^k$$

$$\text{مساحت کاشی} = 3$$

$4^k \bmod 3 = 1$ ⟵ $\frac{4^k}{3}$ باقی‌مانده دارد

یک مربع ۱×۱ صفحه‌ها که حذف شدند ⟵ ۱ (یکی است هرجای باشد)

در این مسئله هم مسئله اصلی یک جای کاشی (۱×۱) خانه کم شد

شکل ۵ کاشی است که می‌بینید مرتب دوتا دوتا صورت که صفحه‌ها خانه کج نیست راست می‌آید

می‌شود.

چون کاشی است شکل ۲ دوتا سطر ۱ پر کرده.

باید مسئله را اینجوری درست کنیم که بتوانیم کاشی‌ها را جابجا کنیم یا نچرخند

اگر بیایم بر اساس کاشی خانه‌ها مسئله اصلی را است خانه‌های مسئله‌ها را حذف کنیم

شرط خانه‌ها این است که یک خانه 2×2 باقی‌مانده بتوانیم کاشی‌ها را بگذاریم که شرط‌ها را

جواب رسمش کنیم

استنکبر مهندسی کامپیوتر دانشگاه صنعتی امیر کبیر
اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

Fill (A, n, Flag)

if $n \leq 2$ ← شرط خاتمه

{

if Flag = BR

A[0][0] = A[0][1] = A[1][0] = counter++;

else if

}

برای اینکه بتوانیم خانه‌های را Flag بدهد.

در هر مرحله آرایه را تا پی می‌کنیم یا در اصل

هنوز است آدرس شروع را برمی‌گردانیم

در غیر اینصورت باید دوباره چهار قسمت تقسیم کنیم

A_1, A_2, A_3, A_4 ← از روی A می‌سازیم

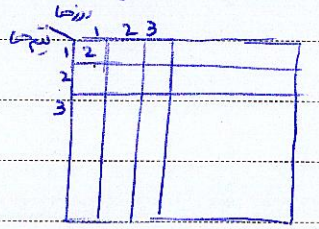
Fill (A_1, $n/2$, BR)

Fill (A_2, $n/2$, BL)

Fill (A_3, $n/2$, TR)

Fill (A_4, $n/2$, TL)

مسئله) جدول مسابقات: n تیم ورزشی می‌خواهند هم با هم مسابقه بدهند. یک روز هر تیم فقط یک مسابقه دارد.

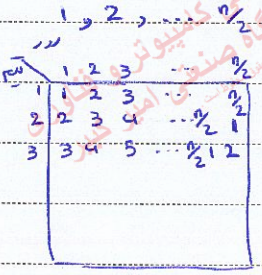


اگر n تیم تقسیم به دو گروه شوند

A B

$n/2$ تیم $n/2$ تیم

هر A ها با هم بازی کنند، تنها با اسامی بازی کرده که در B هستند



1, 2, ..., $n/2$

یک هم چون جدولی برای آن کار کنند

به صورت جدولی پر شود و در یک ردیف یا ستون عملیاتی نداریم.

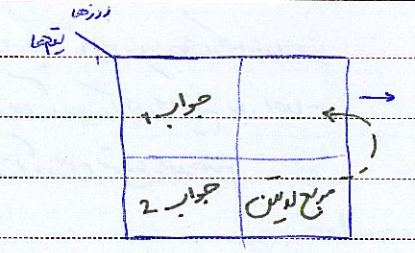
P4PCO

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )



برای بهترین این دوتی ها ابتدا بازی‌ای انجام دهیم
و بعد برنامه‌ریزی کنیم

تیم اول با $n-1$ تیم بازی‌کند

تیم دوم با $n-2$ تیم بازی‌کند

تیم n با 1 تیم بازی‌کند

تعداد کل بازی‌ها (مسابقات) $= \binom{n}{2} = \frac{n(n-1)}{2}$

اگر n زوج باشد در یک روز $\frac{n}{2}$ بازی برگزار می‌شود. می‌توان در $n-1$ روز تیم‌ها را زمان‌بندی کرد

اگر

$1 \longrightarrow n-1$

$\frac{n}{2}$

اگر n فرد باشد یک تیم بازی اضافه می‌شود

در n روز زمان‌بندی می‌شود $n+1-1=n$

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیرکبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

مثال) فرض کنید ۶ تیم داریم می خواهند با هم مسابقه بدهند

به چند تیم تقسیم کنیم

۶ تا سه تیم ← فرد پس یک تیم کاری (ضایعه ای) نیم

تیم ها \ روزها	1	2	3	4	5
1	2	3	4	6	5
2	1	5	3	4	6
3	6	1	2	5	4
4	5	6	1	2	3
5	4	2	6	3	1
6	3	4	5	1	2

+3

۱ ← اضافه است حذف می شود

تیم ها \ روزها	1	2	3
1	2	3	۷
2	1	۷	3
3	۷	1	2
۷	3	2	1

جدول تیم ۷ تایی است می توان ۷ را حذف کرد.

پس از حذف ۷ →

1	2	3	–
2	1	–	3
3	–	1	2

جدول زمان بندی ۳ تیم در جدول اصلی قراری دهیم

یک ستون یا ردیف به ترتیب اضافه می کنیم تا مربع شود مثلا احتیاج داریم جون تیم ها یک روز بیکاری دارند جون سه تیم زوج است می توانیم بازی های بعضی از ستون آخر را در روزهای بیکاری آنها بگذاریم

اگر بخواهیم از زوج ها بازی هایشان را زمان بندی کنیم برای گروه دوم هم به همان ترتیب است و لازم نیست دوباره حساب کنیم فقط باید شماره تیم ها عوض شود زیرا چون گروه ها ۳ تایی است مقدار اول را که برگزیدیم مقادیر ۱ و ۲ و ۳ می کنیم تا گروه دوم بدست آید

دانشگاه صنعتی امیر کبیر
دانشکده مهندسی کامپیوتر و فناوری
واحد علوم و تحقیقات

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	1	4	3	8	5	6	7
3	4	1	2	7	8	5	6
4	3	2	1	6	7	8	5
5	6	7	8	1	2	3	4
6	5	8	7	4	1	2	3
7	8	5	6	3	4	1	2
8	7	6	5	2	3	4	1

i = 8

+4

۱- زیرمسئله اول

۲- زیرمسئله دوم

$T(n) = 2T(n/2) + O(n^2) = \Theta(n^2)$

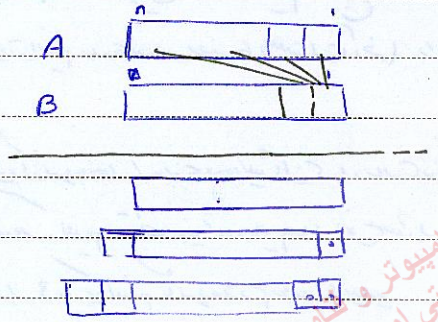
حذف کردن $T(n)$:

$T(n) = O(n^2) + 4T(n/2) + O(1) = \Theta(n^2 \lg^0)$

اگر سادترین حالت زیرمسئله را در ساده‌ترین حالت کپی نکنیم همین طوری که گفتیم هزینه کپی کردن از این بالا می‌رود

$T(n) = O(1) + 4T(n/2) = \Theta(n^2)$

ضرب کردن دو عدد n بیتی:



تعداد ضرب‌های پایه $= \Theta(n^2)$

تعداد جمع‌ها $= \Theta(n^2)$

هزینه اعمال را هم جمع می‌زنیم و در کل

بزرگترین مرتبه پیچیدگی آن n^2 می‌شود

راه‌حل تقسیم: زیرمسئله‌ها همان طوری باشند که هم ضرب‌ها شدنی باشند و تعدادشان کمتر از n^2 شود

ضرب اعداد $n/2$ بیتی ، ... ، ضرب اعداد n بیتی

A | A_1 | A_2

B | B_1 | B_2

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

حالا بیایم ترکیب جمله‌های زیر مسئله را مشخص کنیم

$A \times B = (A_2 + A_1 \times 2^{n/2})(B_2 + B_1 \times 2^{n/2})$

$= A_2 \times B_2 + (A_2 \times B_1 + A_1 \times B_2) \times 2^{n/2} + A_1 \times B_1 \times 2^n$

(1) (2) (3) (4)

تعداد ضرب‌های لازم $M(n) = 4M(n/2)$

از زیر مسئله‌هایی که حدوداً سایز آن‌ها $n/2$ است

$M(n) = \Theta(n^2)$

هزینه ضرب دو عدد n تایی $T(n) = 4T(n/2) + \Theta(n)$

هزینه کل $= \Theta(n^2)$

(جمع جزء سخت)

استفاده از روش تقسیم و حل در حل مسایل خوب است order را پایین می‌آورد ولی هزینه‌های دیگری دارد

(با این مزیت را دارد که ساده‌تر است (شبیه به ساده‌سازی))

یه 4 ضرب داریم که می‌تونیم 3 تا کم کنیم.

$A_1 \times B_2 + A_2 \times B_1 = (A_1 - A_2)(B_2 - B_1) + A_2 B_2 + A_1 B_1$

(1) (2) (3)

و تعداد ضرب 2، 3 حساب شده و دیگر تعداد ضرب‌ها 3 تایی می‌شود

$M(n) = 3M(n/2)$

$= \Theta(n^{\lg 3})$

مرکز تحقیقات کامپیوتری علوم اسلامی
اطلاعات دانشگاه صنعتی امیرکبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

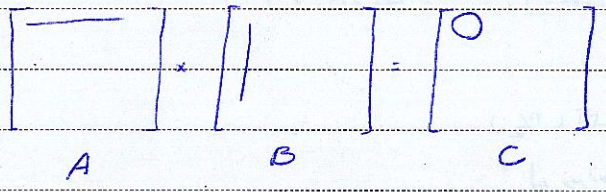
Subject:

Year. Month. Date. ( )

الگوریتم Strassen :

$C_{n\times n} = A_{n\times n} \times B_{n\times n}$

می‌توانیم مستقیم این حاصلضرب را حساب کنیم



هر درایه C از ضرب یک سطر در یک ستون بدست می‌آید که جمعا n ضرب و n-1 جمع لازم دارد

ضرب‌ها $n^2 \times n = n^3$

جمع‌ها $n^2(n-1) = n^3$

بنابراین تقسیم حل می‌تواند این هزینه‌ها را کاهش دهد.

ماتریس‌ها را برای حاصلضرب به ۴ قسمت می‌کنیم

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

برای حل این مسئله از ۷ تا متغیر کمکی استفاده می‌کنیم

$Q_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22})$

$Q_2 = (A_{21} + A_{22}) \times (B_{11})$

$Q_3 = (A_{11}) \times (B_{12} - B_{22})$

$Q_4 = (A_{22}) \times (B_{21} - B_{11})$

$Q_5 = (A_{11} + A_{12}) \times (B_{22})$

$Q_6 = (-A_{11} + A_{21}) \times (B_{11} + B_{12})$

$Q_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22})$

$C_{11} = Q_1 + Q_4 - Q_5 - Q_7$

$C_{12} = Q_3 + Q_5$

$C_{21} = Q_2 + Q_4$

$C_{22} = Q_1 + Q_3 - Q_2 + Q_6$

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

$M(n) = 7M(n/2)$

الگوریتم ضرب ها

$C_{11} = A_{11} \times B_{11} + A_{12} \times B_{21}$

$C_{12} = A_{11} \times B_{12} + A_{12} \times B_{22}$ $\Rightarrow M(n) = 8M(n/2)$

$C_{21} = A_{21} \times B_{11} + A_{22} \times B_{21}$

$C_{22} = A_{21} \times B_{12} + A_{22} \times B_{22}$

Strassen ، 8 ضرب را به 7 ضرب کاهش داده

روش برنامه ریزی پویا Dynamic programming.

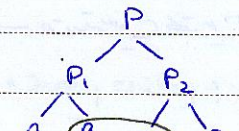
در کدام مسائل به این روش حل می شود

1- مسئله بهینه سازی باشد (min یا max)

2- جواب بهینه مسئله از روی جواب بهینه زیرمسائل برپست آید. (برای اینکه مسئله بهینه حل شود باید زیرمسائل هم بهینه حل شوند) optimal substructure

3- زیرمسائل تکراری زیاد داشته باشد

اجباری نیست.



این زیرمسئله دوبار حل شده (تکراری)

در روش تقسیم و حل B دو بار حل می شد.

ولی در این روش هر زیرمسئله یکبار حل می شود

در این روش ابتدا کوچکترین مسائل را حل می کنیم و جواب آنها را ذخیره می کنیم و به این ترتیب مسائل بزرگتر را حل می کنیم تا به مسئله اصلی برسیم

برای هر مسئله که با این روش حل می شود ذهنی پست پیدا می کنیم که ارتباط بین جواب زیرمسائل را بیان کند. حالت پایه مربوط به کوچکترین مسئله است

باید کوچکترین زیرمسئله را پیدا کنیم و حل آن را شروع کنیم

اطلاعات دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر گروه نرم افزار

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

مثال: شرکت تبلیغاتی بخواهد کار را چنان تقسیم کند که تعداد ضرب ها min شود

$M_1 \times M_2 \times \dots \times M_n$

ضرب ماتریس

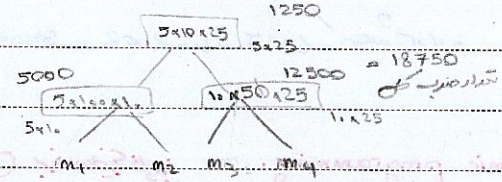
نحوه پرانتزگذاری در تعداد ضرب حاصل تاثیر دارد

$((M_1 \times M_2) \times (M_3 \times M_4))$

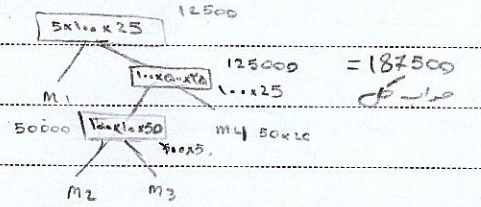
$M_1 \times M_2 \times M_3 \times M_4$

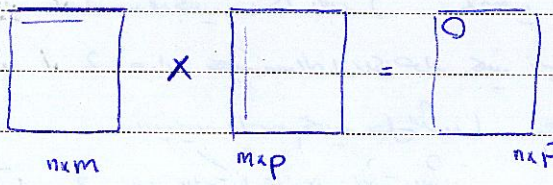
$5\times100 \quad 100\times10 \quad 10\times50 \quad 50\times25$

$(M_1 \times M_2) \times (M_3 \times M_4)$



$M_1 ((M_2 \times M_3) \times M_4)$





تعداد کل ضرب ها $\rightarrow (n \times p) \times m$

برای ضرب n ماتریس بهترین پرانتزگذاری کدام است؟

برای اینکه بتوانیم یک برنامه‌نویسی پویا انجام دهیم ← شرط ① و ② برقرار شد

حال می‌خواهیم تعداد ضرب‌ها min شود ← شرط ①

هر پرانتزگذاری یک درخت ضرب است

آخرین ضرب P

$(\underbrace{\quad}_{P_1}) \times (\underbrace{\quad}_{P_2})$

پرانتزگذاری بهینه

باید پرانتزگذاری P_1 و P_2 هم بهینه باشد

هزینه کل = هزینه P_2 + هزینه P_1 + هزینه ضرب آخر

اگر P_1 و P_2 بهینه حل شده باشند، آنگاه بهینه حاصل از کنار هم گذاشتن آن‌ها

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

$M_1 \times (M_2 \times M_3 \times M_4 \times M_5)$

$\downarrow$

$M_2 \times (M_3 \times M_4 \times M_5)$ و سپس پراندازی همینها را

$\downarrow$

$(M_1 \times M_2) \times (M_3 \times M_4 \times M_5)$

یک زیرمسئله

ست پیاده کنیم

Num. $M_{ij} : M_i \times M_{i+1} \times \dots \times M_j$

بعد $d_{i-1} \times d_i \quad d_i \times d_{i+1} \quad d_{j-1} \times d_j$

این ضرب را خواهیم پراندازی کنیم.

آخرین ضرب

$\downarrow$

$(M_i \times M_{i+1} \times \dots \times M_k) \times (M_{k+1} \times \dots \times M_j)$

$d_{i-1} \times d_i \quad d_{k-1} \times d_k \quad d_k \times d_{k+1} \quad d_{j-1} \times d_j$

مسئله را حل شده فرض کنیم

c_{ij} ⟹ هزینه زیرمسئله‌های چپ

M_{ij} تعداد ضرب‌های چپ در

هزینه ضرب آخر

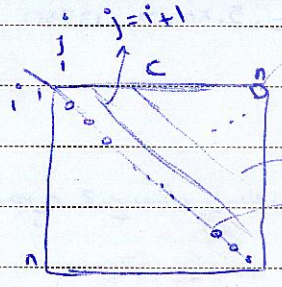
$$c_{ij} = c_{ik} + c_{k+1,j} + d_{i-1} \cdot d_k \cdot d_j \qquad i \le k < j$$

$$c_{ij} = \min_{i \le k < j} \{ c_{ik}, c_{k+1,j} + d_{i-1} \cdot d_k \cdot d_j \}$$ ← جدول نهایی

جدول c دقتا اینسین دارد یک ماتریس برای آن تعریف می‌کنیم

$i = j \longrightarrow c_{ii} = 0$ ... شرط مرزی

یکی از چیزهایی که باید حل شود یک ماتریس داشتیم



c_{ij} را برای مربع‌ها از پایین به بالا حساب می‌کنیم

از سمت راست به چپ

مسئله اصلی $M_{1,n}$ است و جدول $c_{1,n}$ که داریم تعداد ضرب‌های چپ ماتریس

PAPCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد WWW.MAKHFIGAH.COM

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

$M_1 \times (M_2 \times M_3 \times M_4 \times M_5)$

$M_2 \times (M_3 \times M_4 \times M_5)$

$(M_1 \times M_2) \times (M_3 \times M_4 \times M_5)$

یک زیرمسئله — بهینه‌سازی جایگشت را داریم

$M_{ij} : M_i \times M_{i+1} \times \dots \times M_j$

بعد $d_{i-1} \times d_i \quad d_i \times d_{i+1} \quad d_{j-1} \times d_j$

این ضرب‌ها را می‌خواهیم بهینه‌سازی کنیم

آخرین ضرب

مسئله را حل شده فرض می‌کنیم

$(M_i \times M_{i+1} \times \dots \times M_k) \times (M_{k+1} \times \dots \times M_j)$

$d_{i-1} \times d_i \quad d_{k-1} \times d_k \quad d_k \times d_{k+1} \quad d_{j-1} \times d_j$

c_{ij} ⟹ هزینه زیرمسئله‌های بهینه

M_{ij} تعداد ضرب‌های جایگشت در

هزینه ضرب آخر

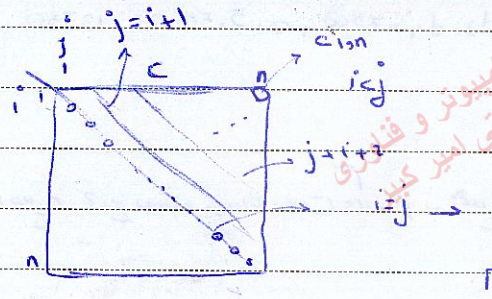
$$c_{ij} = c_{ik} + c_{k+1,j} + d_{i-1} \cdot d_k \cdot d_j \qquad i \le k < j$$

$$c_{ij} = \min_{i \le k < j} \{ c_{ik} + c_{k+1,j} + d_{i-1} \cdot d_k \cdot d_j \}$$ ← بازگشتی

حال ۲ دستاورد داریم یک بهینه برای آن تعریف می‌کنیم

$i = j \rightarrow c_{ii} = 0$ ... شرط مرزی

با کوچکترین زیرمسئله‌ای شروع شد یک بهینه داشتیم



پژوهشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

c_{ij} را برای چه حالاتی از i و j داریم حساب می‌کنیم

از پایین به بالا

مسئله اصلی $M_{1,n}$ است و دنبال $c_{1,n}$ می‌گردیم ← تعداد ضرب‌های بهینه در بهترین

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

دانشکده مهندسی کامپیوتر و فناوری
دانشگاه صنعتی امیرکبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

$M_1 \times (M_2 \times M_3 \times M_4 \times M_5)$

$M_2 \times (M_3 \times M_4 \times M_5)$

$(M_1 \times M_2) \times (M_3 \times M_4 \times M_5)$

یک زیرمسئله — باید برنامه‌ریزی جدید را بهینه بسازیم

$M_{ij} : M_i \times M_{i+1} \times \dots \times M_j$

بعد $d_{i-1} \times d_i \quad d_i \times d_{i+1} \quad d_{j-1} \times d_j$

این ضرب را می‌خواهیم برنامه‌ریزی کنیم

مسئله را حل شده فرض می‌کنیم

آخرین ضرب

$(M_i \times M_{i+1} \times \dots \times M_k) \times (M_{k+1} \times \dots \times M_j)$

$d_{i-1} \times d_i \quad d_{k-1} \times d_k \quad d_k \times d_{k+1} \quad d_{j-1} \times d_j$

c_{ij} ⟹ هزینه زیرمسئله‌های بهینه

M_{ij} تعداد ضرب‌های بهینه در

هزینه ضرب آخر

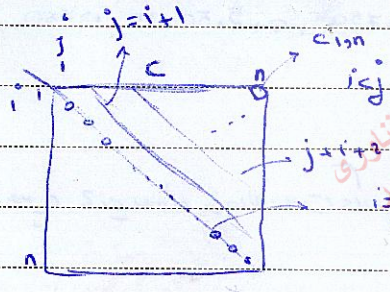
$$c_{ij} = c_{ik} + c_{k+1,j} + d_{i-1} \cdot d_k \cdot d_j \qquad i \le k < j$$

$$c_{ij} = \min_{i \le k < j} \{ c_{ik} + c_{k+1,j} + d_{i-1} \cdot d_k \cdot d_j \}$$ ← زمان خطی

حدس ۲ دقت بررسی دارد ← یک بهترین برای آن تعریف می‌کنیم

$i = j \longrightarrow c_{ii} = 0$ ... شرط مرزی

با کوچکترین زیرمسئله‌ای شروع می‌شود که بهترین را شناخته‌ایم



کپی شده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

c_{ij} را برای همه‌ی iها که $j-i$ برابر است حساب می‌کنیم

از یکی اضافه دارند حساب می‌کنیم

مسئله اصلی $M_{1,n}$ است در نتیجه $c_{1,n}$ را می‌خواهیم ← تعداد ضرب‌های بهینه ماتریس

PAPCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

Subject:

Year. Month. Date. ( )

i \ j	1	2	3	4
1	0	5000 k=1	7500 k=2	13750 k=2
2		0	50000 k=2	137500 k=2
3			0	12500 k=3
4				0

(ج)۲۰

$c_{i,j} = \min\{c_{i,k} + c_{k+1,j} + d_0 \cdot d_k \cdot d_j\}$

$i \le k < j$

$c_{1,2} = c_{1,1} + c_{2,2} + d_0 \cdot d_1 \cdot d_2 = 5\times10\times10$

$c_{2,3} = c_{2,2} + c_{3,3} + d_1 d_2 d_3 = 100\times50 = 50000$

$c_{3,4} = c_{3,3} + c_{4,4} + d_2 \cdot d_3 \cdot d_4 = 10\times50\times25 = 12500$

$M_1 \times M_2 \times M_3 \times M_4$

$5\times100 \quad 100\times10 \quad 10\times50 \quad 50\times25$

$d_0 d_1 \quad d_1 d_2 \quad d_2 d_3 \quad d_3 d_4$

$c_{1,3} = \min$ ($1 \le k < 3$):

$k=1 \rightarrow c_{1,1} + c_{2,3} + d_0 d_1 d_3$

$= 0 + 50000 + 5\times10\times50 = 75000$

$k=2 \rightarrow c_{1,2} + c_{3,3} + d_0 d_2 d_3$

$= 5000 + 0 + 5\times10\times50 = 7500$

$\min = \min(75000, 7500)$

$c_{2,4} = \min$ ($2 \le k < 4$):

$k=2 \rightarrow c_{2,2} + c_{3,4} + d_1 \cdot d_2 \cdot d_4$

$= 0 + 12500 + 100\times10\times25 = 37500$

$k=3 \quad c_{2,3} + c_{4,4} + d_1 d_3 d_4 = 50000 + 0 + 100\times50\times25 = 175000$

$c_{1,4} = \min$ ($1 \le k < 4$):

$k=1 \rightarrow c_{1,1} + c_{2,4} + d_0 \cdot d_1 \cdot d_4$

$= 0 + 37500 + 5\times100\times25 = 50000$

$k=2 \rightarrow c_{1,2} + c_{3,4} + d_0 d_2 d_4 = 5000 + 12500 + 5\times10\times25 = 18750$

$k=3 \rightarrow c_{1,3} + c_{4,4} + d_0 d_3 d_4 = 7500 + 0 + 5\times50\times25 = 13750$

$((M_1 \times M_2) \times M_3) \times M_4)$

PAPCO

WWW.MAKHFIGAH.COM

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

PAPCO

Subject:

Year. Month. Date. ( )

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

برنامه ریزی پویا:

مسأله پیدا کردن طولانی ترین زیردنباله مشترک:

$\vec{a}$: $a_1\ a_2 \dots a_n$

$\vec{b}$: $b_1\ b_2 \dots b_m$

زیردنباله ← انتخاب بعضی از object ها از دنباله اصلی به ترتیب

زیردنباله هر object عضو دنباله اصلی است به ترتیب object ها در دنباله اصلی باشد

a: 1 5 2 3 1

b: 4 1 6 2 1

(طولانی ترین زیردنباله مشترک ← 1, 2, 1)

اگر آخرین object دنباله را هم تا بیشتر کنیم و سعی کنیم زیردنباله ای پیدا کنیم که طولانی ترین باشد آخرین object آن an باشد

Longest Common Subsequence

$a_n = b_m$ اگر ← $LCS(\vec{a}, \vec{b})$ وجود دارد که حرف آخرش $(b_m = a_n)$ باشد

$a_n = b_m \rightarrow LCS(\vec{a}, \vec{b}) = LCS(\vec{a}_{n-1}, \vec{b}_{m-1}) + a_n$ بی اصل b_m را برداشتیم

$a_n \neq b_m \rightarrow$ (حالا می دانیم حرف آخر جواب حتما لازم است)

① $LCS(\vec{a}, \vec{b})$ حرف آخرش a_n ← $LCS(\vec{a}_n, \vec{b}_{m-1})$

② " " b_m ← $LCS(\vec{a}_{n-1}, \vec{b}_m)$

③ " " $\neq b_m, a_n$ ← $LCS(\vec{a}_{n-1}, \vec{b}_{m-1})$

از بین این سه حالت حتما یکی از جواب ها را می دهد

P_{ij}

$\vec{a}: a_1 \dots a_i$

$\vec{b}: b_1 \dots b_j$

L_{ij} ← طول بزرگترین زیردنباله مشترک

$$L_{ij} = \begin{cases} L_{i-1,j-1}+1 & \text{اگر } a_i = b_j \\ \max\{L_{i-1,j-1}, L_{i-1,j}, L_{i,j-1}\} & \text{اگر } a_i \neq b_j \end{cases}$$

b_j جزو جواب نیست

a_i جزو جواب نیست

چون $L_{i-1,j-1} \leq L_{i-1,j}$ و $L_{i-1,j-1} \leq L_{i,j-1}$ ⇒ $L_{ij} = \max\{L_{i-1}, L_{ij-1}\}$

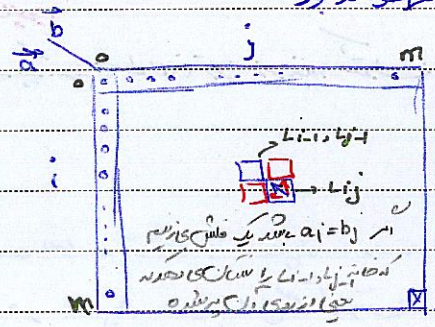
P4PCO

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject.
Year. Month. Date. ( )

اگر یکی از دنباله ها یا هر دو تهی باشد طول زیردنباله مشترک آنها صفر خواهد بود



یک ماتریس $m+1 \times n+1$ برای S_3

حل دنباله ها

از دنباله S_3 راست به چپ به راست ماتریس را پر می کنیم

طول زیردنباله $\rightarrow L_{n,m}$

خانه i,j را با a_i و b_j پر می کنیم اگر $a_i = b_j$ باشد خانه $L_{i-1,j-1}$ را نشانه می کنیم یک جهت می کنیم به $L_{i-1,j-1}$

زیا را برگشته حرکت می کنیم و اگر $a_i \neq b_j$ بود نگاه می کنیم کدام خانه $L_{i-1,j}$ و $L_{i,j-1}$ بزرگتر است max را انتخاب می کنیم

در پر کردن خانه های ماتریس اگر $a_i \neq b_j$ باشد برای قرار دادن عدد به این که کدام max است توجه می کنیم

اگر $L_{i-1,j}$ max بود به سمت چپ و اگر $L_{i,j-1}$ max بود به بالا قرار می دهیم

مثال:

$\vec{a}$: a b a a c d b a

$\vec{b}$: a a b c d

		0	1	2	3	4	5
	0	0	0	0	0	0	0
1		0	↖1	←1	←1	←1	←1
2		0	↑1	↖1	←2	←2	←2
3		0	↖1	↑2	↖2	←2	←2
4		0	↖1	↖2	↑2	↖2	←2
5		0	↑1	↑2	↑2	↖3	←3
6		0	↑1	↑2	↖3	←3	↖4
7		0	↑1	↑2	↑3	←3	↑4
8		0	↖1	↖2	↑3	↑3	↖4

اگر خانه ای مثل خانه 2و3 از طریق max پر شده

هر دو کلی بودن فرقی ندارد کدام را در نظر بگیریم

$b_1 = a_3$

$a_4 = b_2$ $a_5 = b_4$

LCS: a a c d → a_4 a_5 b_5

ساری یادداشت می کنیم

یادداشت می کنیم

$a_6 = b_5$ ← یکسانی رخ داده

حول از آخر به اول می رویم زیردنباله از آخر پر می شود

اندازه بزرگترین ماتریس خانه آخر شامل طول زیردنباله طولانی مشترک است.

برای پیدا کردن خود زیردنباله به جهت فلش ها توجه می کنیم

وقتی به حدهایی برسیم از هر راهی برویم جواب می رسیم برای اینکه مثلاً aa را دارد بخاطر هم فرق نکند

P4PCO 15.9.6 - 15.4.5 - 15.4.1

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

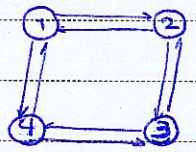
یک مسأله بهینه سازی به شرط دوم را دارد.

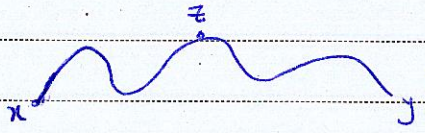
↓ optimal substructure

بنابراین می توان از برنامه ریزی پویا برای حل آن استفاده کرد.

Longest Path

مسأله بهینه سازی حول طول های ماکزیمم





طولانی ترین مسیر بین x و y برای اینکه شرط دوم را داشته باشد باید برابر باشد با جمع طولانی ترین مسیر بین x و z و طولانی ترین مسیر بین z و y که اینطور نیست

③ طولانی ترین مسیر بین ① و ③ : ① → ② → ③

این قسمت طولانی ترین بین ① و ② نیست

② طولانی ترین مسیر بین ① و ② : ① → ④ → ③ → ②

③ و ② " " " : ② → ① → ④ → ③

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

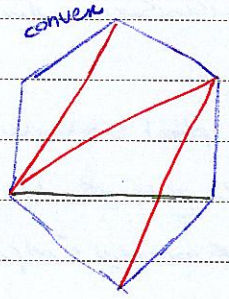
. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

مسأله مثلث بندی بهینه یک چند ضلعی محدب:

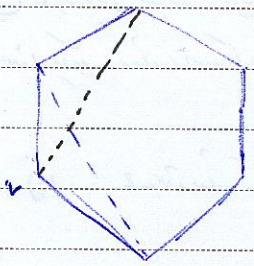


تعریف convex : بعضی از زاویه‌ها کمتر از 180

← قطرهای داخلی را رسم کنیم که بتوانیم مساحت را به چند مثلث تقسیم کنیم (قطرها باید متقاطع نباشند و روی مثلث‌ها روی چند ضلعی باشند) ← تعداد مثلث‌ها ثابت است $(n-2)$

حدس: سیستمی که مجموع طول قطرها min شود

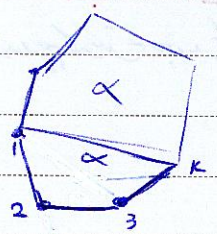
← یک چند ضلعی را به چند روش می‌توان مثلث بندی کرد؟



ابتدا از 1 قطری رد می‌شود یک قطر از 1 رسم می‌شود

← در این حالت می‌توان فرض کرد قطری به راس‌های دور رسم شده که همان ضلع است.

راس دلخواه 1 را در نظر می‌گیریم و یک قطر از آن رسم می‌کنیم که به دو قسمت تبدیل می‌کند. ضرب تعداد حالات مثلث بندی این دو قسمت جواب می‌شود



$$\sum_{k=2}^{n} T(k) \cdot T(n-k+2)$$

این فرمول تعداد حالات تکراری زیاد دارد

اگر بخواهیم سیستمی (تقسیم بهینه) حل کنیم باید از همین روش برویم

عدد کاتالان: $(m_1 + m_2 \times \dots + M_k)(M_{k+1} \times \dots \times M_n)$

$$\sum_{k=1}^{n} T(k) \cdot T(n-k) = \frac{1}{n+1}\binom{2n}{n}$$

$$= \frac{1}{n+1}\binom{2n}{n} \frac{2n!}{n!\,n!} = \frac{(n+1)(n+2)\cdots(2n)}{1\times 2\times 3\times \cdots \times n} \propto \frac{1}{n+1}$$

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

PAPCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

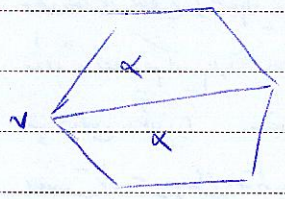
Year. Month. Date. ( )

فکر کنیم) تعداد ساختارهای غیر یکسان برای n چند تاست؟

بینیم optimal substructure برقرار است یا نه؟

یک جواب بهینه داریم برای اینکه جواب کل بهینه باشد

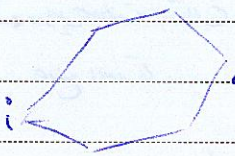
لازم است زیرمسائل هم بهینه باشند ← برنامه سازی پویا



راس دلخواه x را صفر می‌گیریم هر مثلثی که از x رد می‌شود می‌تواند بهینه باشد به دو زیرمسئله تقسیم می‌کنیم جواب‌ها را با هم در ساده‌ترین شکل داریم جواب کل می‌شود جواب زیرمسئله‌ها + مقدار

بزرگترین زیرمسئله یک مثلث است که جواب بهینه‌اش صفر است و بعد برای ۴ ضلعی و بعد برای ۵ ضلعی و بعد... همینطور می‌رویم

وقتی اضلاع زیاد شد ۲ به بیشتر مثلث است (از یک راس شروع می‌شود تا یک راس دیگر ادامه پیدا کند)



مسئله درخت جستجوی دودویی بهینه) optimal Binary Search Tree

یک سری کلمه داریم که متن جدید تکرار شده کلمه‌ها خواهیم جلو بیاریم در dictionary که داریم کلمات چین است invalid باشد

a_i کلمه

P_i احتمال رخداد کلمه در متن

b_i

کلمات مرتب $a_1 < a_2 < \dots < a_n$

کلمه جدید sort شده جدا از حروف الفبا

$a_i < b_i < a_{i+1}$

کلمات نامرتب که در متن هستند می‌شوند invalid

اگر اینجا را search کنیم unsuccessful خارج می‌شود.

کلمات نامرتب از a_0 گرفته تا a_n یا بزرگتر از a_n هستند که $n+1$ دسته تقسیم می‌شوند

P4PCO

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

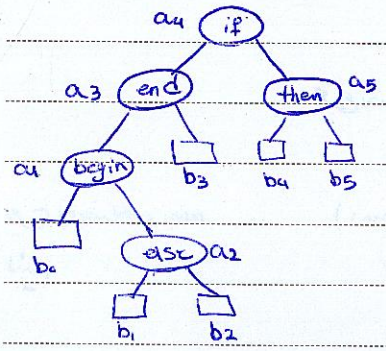
دسته‌ی عبارت با طول n را با $b_0 \; b_1 \; b_{n-1} \; b_n$ نشان می‌دهیم:

$\checkmark$

$\langle a_1 \; a_1 < \langle a_n \; a_{i-1} < a_i < a_n \; \rangle a_n$

q_i ← احتمال رخداد عبارت i ام

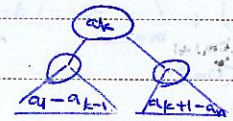
q_0 ← چند کلمه داریم که بین a_i و a_{i+1} قرار بگیرند



برگ‌های نشان‌دهنده‌ی جستجوی ناموفق هستند و برگ‌های داخلی جستجوی موفق

کار ساده این است که ببینیم کدام کلمه ریشه باشد ← همین طوری بین برویم

به ازای هر کلمه که ریشه شد ← subtree های سمت چپ و راست ریشه هایی هستند



جستجوی then : if ، مقایسه‌ی اینم بزرگتر از if پس سمت راست داریم مقایسه‌ی اینم برابر است ← موفق

$$\text{متوسط مقایسه‌ها برای جستجوی موفق} = \sum_{i=1}^{n} p_i \; [\text{depth}(a_i) + 1]$$

برای b_2 ، 4 تا مقایسه داریم ← تعداد مقایسه‌ها برای جستجوی ناموفق عمق آن است

$$\text{متوسط تعداد مقایسه‌ها برای جستجوی ناموفق} = \sum_{i=0}^{n} q_i \cdot \text{depth}(b_i)$$

P4PCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

می‌خواهیم جمع تعداد تست‌های سوق داده شده min شود.

می‌خواهیم یک OBST بسازیم از a_i تا a_j ... $a_i, a_{i+1}, \ldots, a_j$ جوری که شروع می‌شود

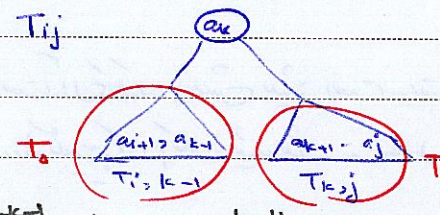
$P_{i+1}, P_{i+2}, \ldots, P_j$

$b_i \quad b_{i+1} \quad b_{j-1} \quad b_j$ ← (طول نادرست بودن)

$q_i \quad q_{i+1} \quad q_{j-1} \quad q_j$

C_{ij} ← جواب مسأله = T_{ij} مجموع تعداد تست‌های اجرا شده بصورت سوق یا نادرست برای

$$C_{ij} = \sum_{m=i+1}^{j} P_m (depth(a_m)+1) + \sum_{m=i}^{j} q_m (depth(b_m))$$ ← این را می‌خواهیم min کنیم



$$C_{ij} = \sum_{m=i+1}^{k-1} P_m \cdot (depth_{T_0}(a_m)+1) + P_k \cdot (depth_{T_0}(a_k)+1) + \sum_{m=k+1}^{j} P_m \cdot (depth_{T_1}(a_m)+1)$$
$$+ \sum_{m=i}^{k-1} q_m \cdot depth_{T_0}(b_m) + \sum_{m=k}^{j} q_m \cdot depth_{T_1}(b_m)$$

$C_{i,k-1}$ $C_{k,j}$

ارتباط عوت دست‌بردهت با عمق اصلی: در درخت اصلی عمق یک a_i کلی بیشتر از عمق آن در زیردرخت است

$$C_{ij} = C_{i,k-1} + C_{k,j} + P_k + \sum_{m=i+1}^{k-1} P_m + \sum_{m=k+1}^{j} P_m + \sum_{m=i}^{k-1} q_m + \sum_{m=k}^{j} q_m$$

$$= C_{i,k-1} + C_{k,j} + \underbrace{\sum_{m=i+1}^{j} (P_m + q_m) + q_i}_{= w_{ij}}$$

با ذل حسابی داریم k

$$= C_{i,k-1} + C_{k,j} + w_{ij}$$

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر

P4PCO

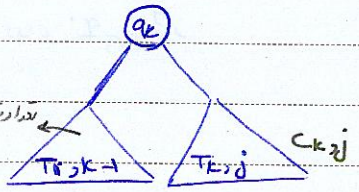
. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

ادامه درخت دودویی جستجوی بهینه : OBST

بعد از مرتب کردن کلمات تعداد زیادی درخت می‌توان از روی آن ساخت و همین طور برای زیر درخت‌ها هم می‌توان هر node را به عنوان ریشه در نظر گرفت.



مرتب ← $a_{i+1} < \dots < a_{k-1}$ $a_{k+1} < \dots < a_j$

نامرتب $b_i < \dots < b_{k-1}$ $b_k < \dots < b_j$

$$c_{ij} = \sum_{m=i+1}^{j} p_m \cdot (\text{depth}(a_m)+1) + \sum_{m=i}^{j} q_m \cdot \text{depth}(b_m)$$

این $\sum$ را به سمت راست ببریم — این b را به درون

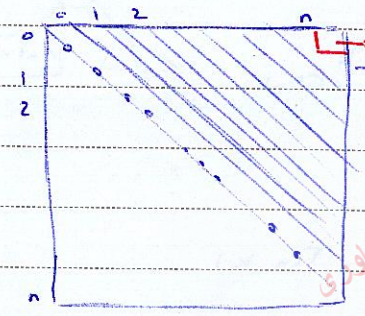
← هر node در زیر درخت آن عمقش h باشد در درخت اصلی $h+1$ است

$c_{ij} = c_{i,k-1} + c_{kj} + w_{ij}$ ← وقتی که جواب بهینه با ریشه a_k است

باید به ازای اینکه هر node ریشه باشد حساب کنیم و بعد min بگیریم

$$c_{ij} = \min_{i<k\le j}(c_{i,k-1} + c_{kj}) + w_{ij}$$

به ازای تمام مقادیر k حساب کنیم و به ازای هر k که min شد آن k را یک جایی save کنیم



برای نگهداری r_{ij} ها

جواب کل مسئله در این خانه است

به جدولی که شروع به پر کردن این می‌کنیم

پرحجم‌ترین زیر مسئله

چون اصلا مقایسه‌ای انجام نمی‌شود ← T_{ij} ← درخت بهینه ← از اینجا به $c_{ij}=0$

اما در این هیچی نیست و k هیچ مقداری نمی‌تواند داشته باشد

حداقل این که مسایل بزرگ را حل کردیم از رابطه بازگشتی جواب مسایل کوچکتر را بدست می‌آوریم

P4PCO

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

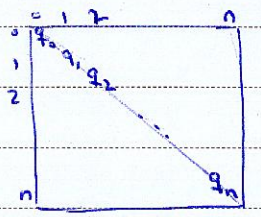
Subject:

Year. Month. Date. ( )

$w_{ij} = \sum_{m=i+1}^{j} (p_m + q_m) + q_i = w_{i,j-1} + p_j + q_j$ ← می‌توانیم جمله آخر را جدا کنیم

به همین گونه می‌توانیم از پایین حساب کنیم w_{ij} را

$w_{i,i} = q_i$



قطر اصلی ماتریس w_{ij} ها q_i می‌شود

مثال (OBST)

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

$a_1 < a_2 < a_3 < a_4 < a_5$

p_i ها برای یک کلمه

q_i برای کلمه‌های رد شده از کلمه

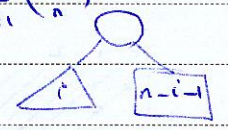
P_1 P_2 P_3 P_4 P_5 ← جدول a_1 تا a_5 مرتب شده

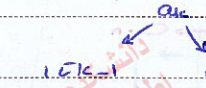
کل حالتهایی که دیتا جستجوی جزء این 5 تا نیست دارد q_0 هست ← b_0 احتمال آن

در کل 5 تا کلمه valid داریم و 6 تا حالت invalid که احتمال آنها در جدول فوق است

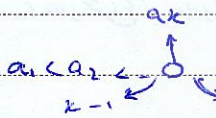
روی n تا کلمه چند تا درخت جستجوی دودویی می‌توان ساخت؟

تعداد درخت‌های جستجوی دودویی با n گره: $N \sum_{i=1}^{n} T(i) \cdot T(n-i-1) = \frac{1}{n+1}\binom{2n}{n}$





این برای وقتی است که درخت ما ترجیحی نداشته باشد (جستجوی دودویی نباشد)



$T(n) = \sum_{k=1}^{n} T(k-1) \cdot T(n-k)$

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

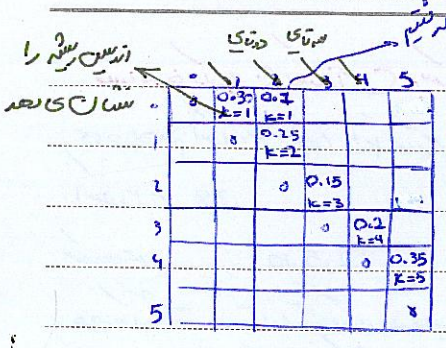
P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )



	0	1	2	3	4	5
0	0	0.3 k=1	0.1 k=1			
1		0	0.25 k=2			
2			0	0.15 k=3		
3				0	0.2 k=4	
4					0	0.35 k=5
5						0

تعریف بازگشتی تقریبی اصلی برنامه پویا برای مسئله ما

$$C_{0,1} = \min_{0<k\le 1}\{c_{0,0}+C_{1,1}\} + w_{0,1} = w_{0,0}+P_1+q_1$$

$k=1$ فقط

$$C_{1,2} = \min_{1<k\le 2}\{c_{1,1}+C_{2,2}\} + w_{1,2} = w_{1,1}+P_2+q_2$$

$k=2$ فقط

$$w_{2,3} = w_{2,2}+P_3-q_3$$

	0	1	2	3	4	5
0	0.03	0.03	0.05			
1		0.10	0.25			
2			0.05	0.15		
3				0.05	0.2	
4					0.05	0.35
5						0.10

$$C_{0,2} = \min_{0<k\le 2} \begin{cases} k=1 \Rightarrow c_{0,0}^{0} + C_{1,2}^{0.25} + w_{0,2}^{0.45} = 0.70 \\ k=2 \Rightarrow c_{0,1}^{0.30} + C_{2,2}^{0} + w_{0,2}^{0.45} = 0.75 \end{cases} \quad \text{min}$$

$$w_{0,2} = w_{0,1}+P_2+q_2 = 0.30+0.15+0.45$$

$a_1, a_2 \Rightarrow$ a_1 a_2

روش حریصانه Greedy

برای حل مسایل بهینه سازی زیر مسایل به صورت بهینه حل می شوند.

شبیه برنامه ریزی پویا ـ بین چند تا انتخاب باید min یا max رو پیدا کنیم ولی در حریصانه با تجزیه و تحلیل مسئله می بینیم بهینه است یا نه، آنرا انتخاب می کنیم و جلو می رویم

انتخاب حریصانه (greedy Choice) ← سعی می کنیم بهینه محلی رو پیدا کنیم به راه حل بهینه برسیم

activity Selection مسائلی که با روش برنامه ریزی پویا حل میشوند حل است با روش حریصانه حل شود

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

مساله خرد کردن اسکناس به سکه ها

می خواهیم اسکناس ها را با کمترین تعداد سکه خرد کنیم. کمترین تعداد سکه معادل اسکناس را پیدا کنیم

اسکناس 28

سکه های 1,2,5,10

هر چه سکه بزرگتر باشد تعداد کمتری سکه خواهیم داشت.

انتخاب حریصانه: سکه های بزرگتر اول انتخاب شوند

اول سکه ای بزرگتر است 10 را برمی داریم در 28 حداکثر 10 است پس 8 تومان باقی می ماند برای 5 تومنی داریم

3 تومن باقی می ماند سکه دیگری نداریم که داریم

$28 = 2\times10 + 1\times5 + 1\times2 + 1\times1$

5 سکه

تعداد سکه های کمتری نمی توان پس اسکناس را خرد کرد پس جواب بهینه است.

اسکناس 15

سکه 1,5,11

انتخاب حریصانه ← 5 سکه ⇒ $15 = 1\times11 + 4\times1$

انتخاب بهینه ← 3 سکه ⇒ $15 = 3\times5$

پس ثابت است انتخاب حریصانه تجزیه انتخاب بهینه نشود

تمرین) نشان دهید اگر سکه ها توان دلخواهی از یک متناوبی باشند ...

برای این مساله انتخاب حریصانه جواب بهینه است.

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

مسأله زمان بندی کارها:

یک پردازنده داریم و تعدادی برنامه که قرار است در پردازنده اجرا شوند

مجموعه کار $\{T_i\}$

زمان پردازش $\{t_i\}$

cpu

Task را به چه ترتیبی به cpu بدهیم که اجرا شدن تا زمان پایان میانگین min شود

← پیدا کردن ترتیب اجرای کارهایی پردازنده است به طوری که زمان پایان گیری میانگین کمینه شود

زیرا که برنامه وارد کار بیشتر شود تا جواب دادن برنامه

فاصله بین تحویل کار و اتمام آن = زمان پایان کار

T_1, T_2 → مجموع زمان پردازش $= t_1 + t_2$

t_1, t_2 → پایان کار $= t_1 + t_1 + t_2 = 2t_1 + t_2$

اگر T_1 اول اجرا شود بعد T_2 ← زمان پایان T_1 برابر است با t_1 و زمان پایان T_2 برابر $t_1 + t_2$ حال به کار اول عمل شود

$T_1, T_2, \ldots, T_n$ ⇒ اگر با این ترتیب اجرا شوند زمان پایان = ؟

T_1 زمان پایان t_1

T_2 " " $t_1 + t_2$

T_n " " $t_1 + t_2 + \ldots + t_n$

مجموع زمان پایان ها: $\sum_{i=0}^{n} (n-i+1)\, t_i$

برای min شدن ← کارهایی که ضریب بیشتری دارند باید از همه کمتر باشند

انتخاب حریصانه:

کارها را بر اساس ترتیب صعودی زمان پردازش اجرا شوند

کسی که کمترین اول بخورد کسی که یک ذره بیشتر است اوف

می توانیم اثبات کنیم که بهترین انتخاب همین است.

دو تا ترتیب در نظر می گیریم

هر دو یکی است

① ترتیب $T_i, \ldots, T_a, T_b, \ldots, T_j$

② ترتیب $T_i, \ldots, T_b, T_a, \ldots, T_j$

وقتی $t_a < t_b$

دانشکده مهندسی علوم کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

اگر ثابت کنیم که ① بهتر از ② است بعد از این ترتیب عین هم هستند فقط جای T_a و T_b عوض شده. آنها نشان دهیم زمان پردازش ① کمتر از ② است ثابت کردیم. برای هر دو task دلخواه نشان دهیم و تعمیم دهیم به یک ترتیب صعودی می‌رسیم.

زمان پاسخ‌ها را با هم جمع می‌زنیم

زمان پاسخ T_a تا قبل از T_a ← α ← مجموع زمان پاسخ‌های نه‌ها از ابتدا تا T_a

← برای هر دو یکی است چون ترتیبشان یکی است.

T_1 مجموع زمان پردازش کارها از ابتدا تا T_a

مجموع زمان پاسخ‌های نه‌ها از ابتدا تا T_b ← α

T_1 مجموع زمان پردازش کارها از ابتدا تا T_b

T_2 مجموع زمان پردازش‌ها کارها بعد از T_b تا آخر

" " " " T_a "

← برای این دو T_2 برابر شد برای دو هم T_2 خواهد بود

β زمان پاسخ کارها ← برای هر دو

① مجموع زمان پاسخ ترتیب $= \alpha+\beta+(T_1+t_a)+(T_1+t_a+t_b) = \alpha+\beta+2T_1+t_a+t_b+t_a$

زمان پاسخ T_a ← ، زمان پاسخ T_b ←

② مجموع زمان پاسخ ترتیب $= \alpha+\beta+(T_1+t_b)+(T_1+t_b+t_a) = \alpha+\beta+2T_1+t_b+t_a+t_b$

این عین هم هستند فقط در جمله آخر تفاوت دارند

زمان پاسخ ترتیب ② > زمان پاسخ ترتیب ① $\Rightarrow$ چون $t_a<t_b$

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

PAPCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

مسالم انتخاب فعالیت ها: activity Selection

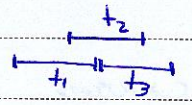
مجموعه فعالیت $\{a_i(s_i, f_i)\}$ — s_i: شروع، f_i: خاتمه

هدف: انتخاب بیشترین تعداد فعالیت که تضادی با هم ندارند

محور زمان ⟶

انتخاب حریصانه ①: فعالیت های کوتاه تر اول انتخاب شوند

مثال نقض:



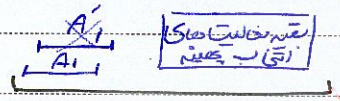
② انتخاب حریصانه بهینه: انتخاب فعالیتی که از همه زودتر تمام می شود و حذف تمام فعالیت هایی که با این تضاد دارند.
برای انتخاب بعدی آنهایی که با این تضاد دارند از دور خارج می شوند و دوباره همین ترتیب) و ادامه انتخاب می کنیم
⟵ انتخاب فعالیتی که کمترین زمان خالی را دارد و حذف تمام فعالیت های متضاد و ادامه به همین صورت

③ انتخاب فعالیتی که کمترین تضاد را دارد.

حالا نشان می دهیم که در بین انتخاب بهینه است، برای این کار:

قضیه ← با فرض اینکه فعالیت A_1 دارای کمترین زمان خاتمه است - ثابت می کنیم که یک انتخاب بهینه از فعالیت ها وجود دارد که A_1 در این فعالیت ها باشد

برهان: فرض کنید که یک انتخاب بهینه از فعالیت هایی که A'_1 در این فعالیت هاست:



A'_1 را حذف می کنیم و A_1 را به آن اضافه می کنیم

چون می دانیم $f_{A_1} \leq f_{A'_1}$

پس با هیچ تضاد نخواهد داشت می توانیم اضافه کنیم و چون بقیه انتخاب ها بهینه هستند پس این انتخاب هم بهینه است پس همیشه یک فعالیت انتخاب بهینه داریم که با A_1 شروع می شود

انتخاب A_1 مجبور از دست دادن جواب های بهینه بقیه فعالیت ها نخواهد شد.

P4PCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Month. Date. ( )

الگوریتم حریصانه انتخاب فعالیت‌ها:

1. مرتب کردن فعالیت‌ها بر اساس زمان خاتمه به صورت صعودی
2. انتخاب فعالیت با زمان خاتمه کمینه
3. حذف فعالیت‌های متضاد
4. ادامه از 2

این طور آنقدر ادامه می‌دهیم تا تمام شود

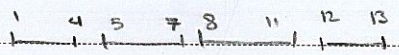
شماره فعالیت (مثال)	S_i	f_i	
1	1	4	✓
2	3	5	①
3	0	6	①
4	5	7	✓
5	3	8	①
6	5	9	②
7	6	10	②
8	8	11	✓
9	8	12	③
10	2	13	①
11	12	13	✓

بر اساس زمان خاتمه مرتب شده

فعالیتی که زمان start آن کمتر است را انتخاب می‌کنیم

از بین فعالیت‌هایی که min است انتخاب می‌کنیم.

در بره متضادها را حذف می‌کنیم.



دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر

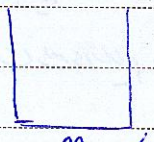
PAPCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

یک نسخه از کوله پشتی:

یک کوله پشتی داریم که حداکثر گنجایش m است و یک تعداد شی داریم که هرکدام وزنی دارد می خواهیم روی کوله پشتی قرار دهیم که این شی ها باید دزشان حجما شود و از بین اشیای داخل کوله پشتی max شود.

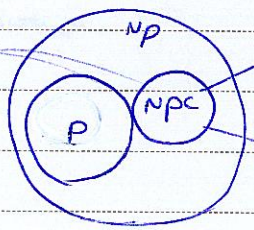


$\{O_i(w_i, v_i)\}$ مجموعه اشیاء
وزن ها ارزش

$\sum w_i \leq M$
براساس اشیا داخل کوله پشتی

$\text{Max} \sum v_i$

np-complete
بدرستی حریصانه حل می شود ← یک حل بهینه داریم یا اصلا بهینه ای داریم ← ۰-۲
انتخاب حل بهینه یا هیچ برنامه سوزی پویا (زبان چند جمله ای)
این مسئله دو نسخه بصورت دارد
بهینه ای پارا متر داخل است که همان است تعداد اشیاء هست $O(nm)$
مسئله های بالقوه
اجازه داریم کسری از یک شی را انتخاب کنیم (fractional) کسری
$m = 2^n$ ← P ← چند جمله ای زبان چند جمله ای
class
می توان بدرستی حریصانه حل کرد
یک تابع هزینه با ورودی و خروجی مشخص
شبه زمان حل با برنامه سوزی پویا



strongly NPC قوی
ضعیف ← برنامه سوزی پویا
بهترین این دو برای ما بحلوم نیست که آیا اصلا راه حل بهینه دارد یا نه؟

بهتر است شی ای که وزن کمتر ارزش بیشتری دارد را انتخاب کنیم
این باید بیشتر باشد $\frac{v_i}{w_i}$
$O_i : \frac{v_i}{w_i}$ ← انتخاب حریصانه

انتخاب حریصانه: انتخاب اشیا حسب ترتیب نزولی $\frac{v_i}{w_i}$ تا اینکه کوله پشتی پر شود.

این صفحه توسط مرکز تحقیقات کامپیوتری علوم اسلامی دانشگاه صنعتی شریف تایپ گردید
اطلاعات دانشگاه صنعتی شریف

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

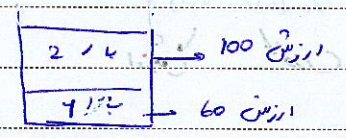
اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

برای مثال Fractional

ارزش هر کیلو	ارزش	وزن	بسته ها
6	60	10	1
5	100	20	2
4	120	30	3

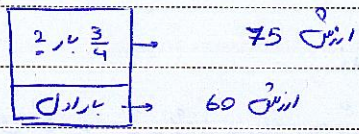
مثال Fractional را حل می‌کنیم

ابتدا به ارزش بیشتر است را برای داریم ...



M=30

مجموع ارزش‌ها = 160



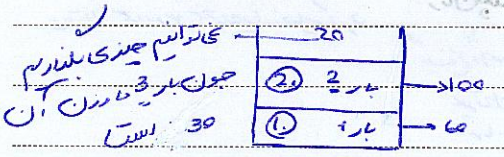
M=25

مجموع ارزش‌ها = 135

برای مثال 1-0

می خواهیم ببینیم ارزش چه مقدار است

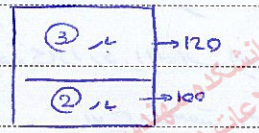
جواب را می‌بینیم



M=50

⇒ ارزش کل = 160

جواب بهینه :



⇒ ارزش کل = 220

پس با چنین روشی می‌توان مثال 1-0 کوله پشتی را حل کرد

: Fractional

کوله را یک تعداد واحد وزن پر می‌کنیم چون Fractional است می‌توانیم هر کدام را به تعداد دلخواه وزن تقسیم می‌کنیم زیرا کوله را با واحد وزنی که ارزش بیشتری دارد پر می‌کنیم به همین است که انتخاب بهینه است

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

چرا برای 0.6 جواب نمی دهد؟ یا چرا برای 0.5 جواب می دهد؟

زمانی که M بزرگ است Bit بیشتری را تقسیم کنیم جواب بهتری می دهد.
به این علت که می توانیم باور را تقسیم کنیم مقداری از فضایی که در کوله پشتی را تشکیل می دهد کمتر است این مقدار هم در درست کردن انتخاب های قبلی می شود و در نتیجه هم بیشتر می شود پس برای 0.6 جواب می دهد.

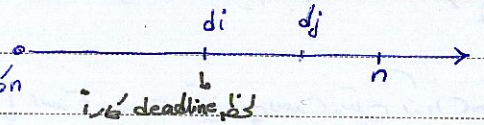
مسأله زمان بندی کارها با جریمه تأخیر

اگر از Deadline گذشت جریمه می شویم ← مجموع جریمه ها min شود.

مجموعه کارها { Ti (ti , di , wi) }
ti: زمان انجام
di: deadline
wi: جریمه

فرض: ti = 1 ms
هر کارها یک اندازه زمان نیاز دارند

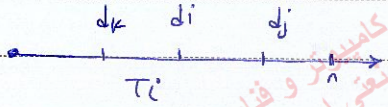
wi → میزان جریمه پرداخت زمان نشده از deadline بستگی دارد ← ثابت



کار
کل deadline ها

زمان بندی کارها در n واحد زمان اجرا می شود

وقتی کاری هست جریمه تعلق می گیرد بهتر است آخر قرار گیرد چونکه هم جریمه تعلق نگرفته هم اینکه هر چقدر از deadline آن گذشت برمیزان جریمه تأثیری ندارد. اگر آخر قرار ندهیم ممکن است بعد از هم جریمه تعلق بگیرد که قبلش خالی باشد قبل از deadline می توانیم زا اجرا دهیم



بهترین کار را با فاصله قبل از deadline اجرا دهیم ← برای کارهایی deadline آخر قبل از di است

وقتی داشته باشیم

منطقی است که کارها را به ترتیب نزولی جریمه هایشان اجرا دهیم.

PAPCO
دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

onth. Date. ( )

۱- کارها به ترتیب نزولی جریمه ها مرتب می‌شوند

2- هر کار حتماً قبل از deadline زمان بندی می‌شود

3- اگر محل مورد نظر قبلاً اشغال شده بود به جلو حرکت کرده و در اولین جای خالی زمان بندی می‌شود

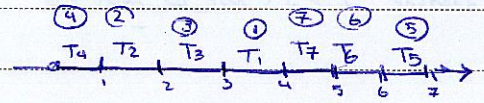
4- اگر هیچ محل خالی قبل از deadline آن موجود نباشد این کار جریمه متوقع می‌شود و به انتها زمان بندی می‌رود.

سپس اگر هر کار deadline متمایز باشد هلی در جریمه خواهی می‌شود و کاری پیچیده ای ندارد.

سپس اگر چند کار deadline یکسان داشته باشند حالتی متناسب با خاصیت deadline

مثال)

کارها	1	2	3	4	5	6	7
w_i	70	60	50	40	30	20	10
d_i	4	2	4	3	1	4	6



اول اینکه چه چیزی را زمان بندی کنیم ⇐ قبل از @

برای T3 چون از 4 به راست یکی عقب تر رسیم قبل از 3 ذخیره می‌کنیم

6 و 5 ، برسیم 1 است چون قبل از راست قبل از آن هم پر شده بنابراین جریمه داری می‌شود.

جریمه = 30 + 20 = 50

پیچیدگی زمانی $O(n \log n)$

$O(n^2 + n\lg n) = O(n^2)$

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

مسأله درختواره بهینه:

تعداد بیت مورد نیاز برای ذخیره کاراکتر

$$\min \sum_{i=1}^{n} P(c_i) \times d(i)$$

تعداد بیتی که در ستون سمت راست نوشته شده

می خواهیم این مجموع را min کنیم

در ASCII همیشه تعداد بیت برابر با 8 می باشد (ادی که در خاستن بیشتری دارد کوتاه تری به آن اختصاص یابد)

الگوریتم حاضر:

برچسب های این درخت را در دفتر مشخص می خواهیم

برایشان کد تولید کنیم

برچسب های این درخت را چپ به راست می خوانیم:

کد هر برگ برچسب یال ها از ریشه تا برگ

کد a: 00

کد هر حرف: برچسب یال ها مسیر ریشه تا حرف (برگ)

a ... ①
b ... ②

①، ② نباید هم پیشوند شوند

حرفی که طولش کوچکتر است نباید پیشوند کدی حرف باشد که طولش بیشتر است

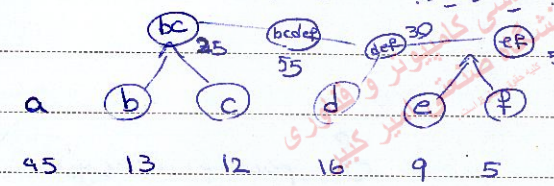
داشتن این خاصیت: خاصیت پیشوندی

پس کد با طول متغیر باید خاصیت پیشوندی داشته باشد.

این کد حاضر چون حرف در برگ ها است پس خاصیت پیشوندی نداریم.

طول کد = طول مسیر از ریشه تا برگ پس هر چه درخت بلندتر باشد کد کوتاه تری خواهد داشت.

مثال)



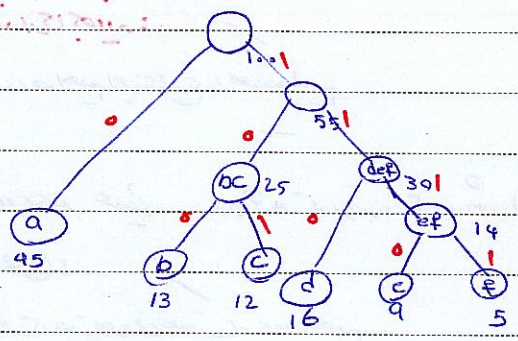
اول دو حرفی که کمترین فرکانس را دارند با هم merge می شوند و حرف جدید با فرکانس مجموع آنها به جدول recursive اول درختی که با هم min هستند را با هم merge می کنیم...

PAPCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )



a : 0

b : 100

c : 101

d : 110

e : 1110

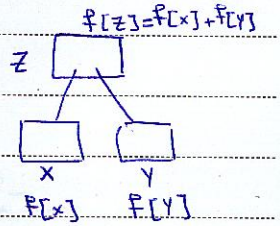
f : 1111

طول نمایش با کد اسکی = 100×3=300 بیت

با کد فشرده = 45×1+13×3+12×3+16×3+9×4+5×4 = 224 بیت

الگوریتم:

```
Huffman (C)
{
  n ← |C|
  Q ← C                    صف اولیت‌دار برحسب ورودی برحسب فرکانس
  for i=1 to n-1 do
    z ← allocate_node()
    x ← left[z] ← Extractmin(Q)
    y ← right[z] ← Extractmin(Q)
    f[z] ← f[x]+f[y]
    Insert(Q, z)
```



دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

دو را از صف برمی‌داریم و جایش

z را برمی‌گردانیم

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

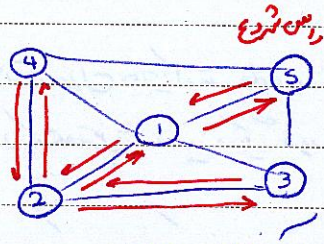
الگوریتم های گراف:

الگوریتم های پیمایش گراف

(Depth First Search) DFS

(Breadth First Search) BFS

DFS:



راس شروع

اگر یک یال هم بندی به یک node visit شده باشد

اگر همه node ها ویزیت شده هم بندی است یک node

برای اینکه یعنی هم بندی دیگر گراف نگیریم

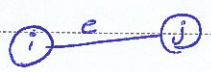
ی درخت جدید بین بست برسیم بعد از آن یک مرحله بالای بیم و از آن ادامه می دهیم

برای گراف هم بند

```
DFS(v)
Begin
  mark[v] = visited        node v بیست همسایه
  for each w in L(v) do
  if mark[w] = unvisited then
    DFS(w)
  end

Depth First Search
for v=1 to n do
  if mark[v] = unvisited then
    DFS(v)
end
```

هر edge در گراف دوبار بررسی می شود یکی برای اینکه ببینیم که آیا از همسایه ویزیت نشده ای هست یا نه دیگری هم برعکس حالی می کند



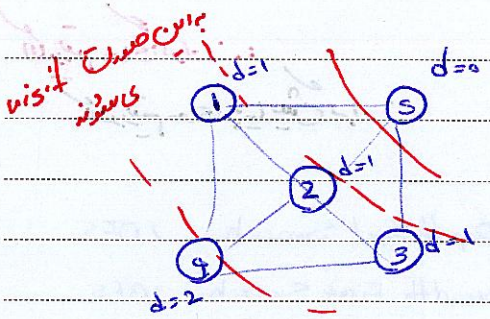
DFS زمان $\Theta(|V|+|E|)$

P4PCO
دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )



: BFS

مثل الگوریتم درخت است.

ابتدا node شروع حساب در مرحله دوم تمام همسایه‌های current node را visit می‌کنیم

اگر d نشان‌دهنده تعداد یال‌هایی که start node به node مورد نظر است در واقع طول کوتاه‌ترین مسیر از راس شروع است تا راسی که کوتاه‌ترین مسیر در یک گراف بدون وزن)

هر node را یک بار برای همسایه‌های آن را هم یک بار بررسی می‌کنیم ← هر یال دو بار بررسی می‌شود

$O(|V|+|E|)$

یک Queue درست می‌کنیم که node را در صف می‌گذارد یک node برای دارد همسایه‌های visit نشده را به صف قرار می‌دهد و یکی از صف بیرون می‌آورد.

Q: ② ① ③ ④ ⑤

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

الگوریتم های کوتاه ترین مسیر (Shortest Path)

$G(V,E)$ یال‌ها

$\min \sum w(e_i)$ ← min جمع یال‌های مجموعه مسیر

$e_i \in path(u,v)$ وزن

الگوریتم ← Dijkstra (حریصانه)

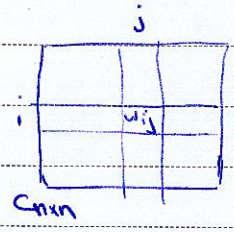
Floy_warshall (برنامه سازی پویا)

الگوریتم Dijkstra

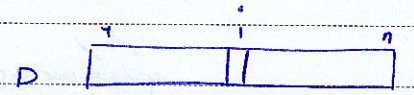
همسایگی

فرض می‌کنند ماتریس وزن‌ها به صورت ماتریس مجاورت است.

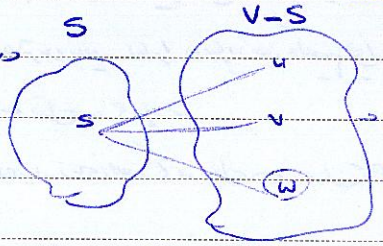
Single Source یک مبدأ دارد (source node) و مسیرهای بهینه



یک آرایه یک بعدی در نظر می‌گیریم



این الگوریتم D را در مرحله به مرحله می‌سازد و هر بار جلو می‌رود



در ابتدا فقط مبدأ درونش است

از اینجا w، min است پس w از V-S برای S

می‌آید به S منتقل می‌کنیم

مرحله اول ماتریس D را از روی C پر می‌کنیم. فرض می‌کنیم گره شروع گره یک باشد

$D[i] = C[1,i]$ ← وزن یالی که از راس 1 وصل می‌کند

اگر بخواهیم فقط از یک یال استفاده کنیم (مسیر بسازیم) یا جواب را دارد (∞)

وگرنه از اینجا کمتر نبود، زیرا به عنوان جواب انتخاب می‌کنیم

بین این دو min می‌گیریم

$D[v]$

$D[w] + C[w,v]$

P4PCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

Dijkstra

begin

S={1};

for i=2 to n do

D[i]=c[1,i];

for i=1 to n-1 do O(n)

begin

choose a vertex w in V-S

such that D[w] is minimum

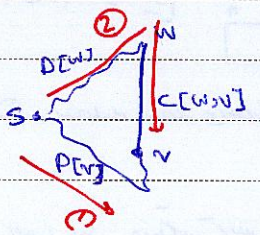
S=S∪{w}

for each vertex v in V-S do O(n)

D[v]=min{D[v], D[w], c[w,v]}

برای کل الگوریتم

$O(n^2)$ <=



سه تا از رئوس را به ترتیب به زبان ها می گیریم که

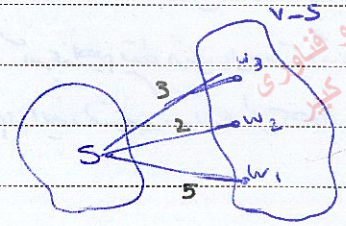
min بودن به عنوان مسیر اصلی انتخاب می شود

از مسیر کوتاه تر بودن به ما را جای ذخیره کنیم می آید

بدانیم node ها قبل یا چی نبوده

انتخاب جدید زمانی است که یال ها زدن صفر یا منفی نداشته باشند

در هر مرحله که برای پیدا کنیم می توانیم دفعه تر مسیری پیدا کنیم که به جدید node های بیشتر مسیر کوتاه تری پیدا کنیم



w2 کوتاه ترین فاصله را از S دارد و هیچ مسیر دیگری

پیدا نمی کنیم که کوتاه تر باشد چون یال منفی

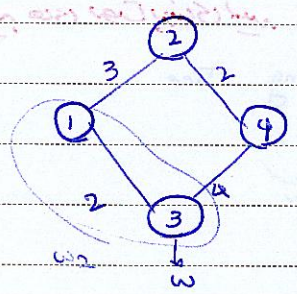
یال های دیگر را هم به آن اضافه نمی کنیم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر
P4PCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید .

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )



	1	2	3	4
① مرحله → D	-	3	2	∞
② مرحله →	-	3	-	6
			w	
③ مرحله →	-	-	-	5

کم هزینه ترین مسیری که پیدا کردیم از قبل کم هزینه تر بوده ← update ای نیستیم

الگوریتم Floyd_warshall

All pairs کم هزینه ترین مسیرها بین هر زوج رأس دلخواه

Floyd_warshall

begin

for i=1 to n do

A[i,j]=C[i,j]

for k=1 to n do

for i=1 to n do

for j=1 to n do

if A[i,k]+A[k,j]<A[i,j] then

A[i,j]=A[i,k]+A[k,j]

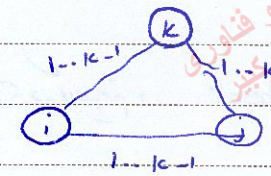
end

وزن یال ها Cn×n

An×n → A[i,j] طول کم هزینه ترین مسیر از i به j

در صورتی که update می شود تعداد عمل

$\Theta(n^3)$



دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر

PAPCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

الگوریتم های درخت پوشای کمینه

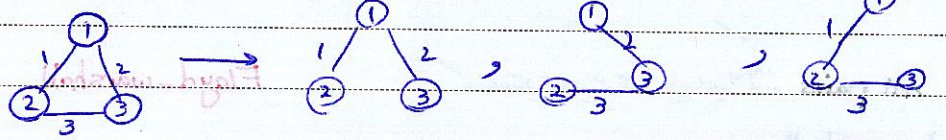
(MST) minimum spanning Tree

G(V,E)

گراف ← درخت

↙ پوشا (شامل همه رئوس)

↘ کمینه (مجموع وزن یال ها کمینه باشد)



3 5 4

درخت پوشای کمینه

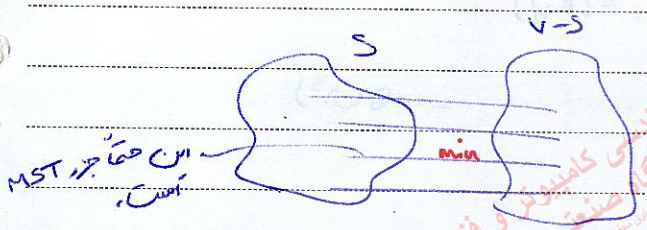
ای کل رئوس را دارد.

الگوریتم ← prime

↙ kruskal

هر دو حریصانه هستند و انتخاب greedy آن بر اساس یک قضیه است.

قضیه: اگر رئوس گراف G(V,E) را به دو قسمت S و V-S تقسیم کنیم یال با مینیمم کمینه بین S و V-S متعلق به MST است.



اگر هم نباشد حداقل یک یال دیگر دارد.

تفاوت هم این دو الگوریتم در subset است

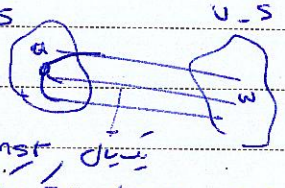
دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیر کبیر
P4PCO
به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:
Year. Month. Date. ( )

الگوریتم prime

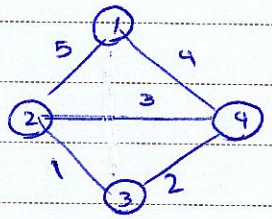
یک رئوس دلخواه را انتخاب می‌کنیم و در مجموعه S قرار می‌دهیم
سپس S و V-S آن یالی که مینیمم هزینه را پیدا کرده و مرتبط
آن در V-S یعنی u را به S اضافه می‌کنیم و این کار
را ادامه می‌دهیم. (این کار را n-1 بار انجام می‌دهیم)



یال‌هایی که انتخاب می‌شوند یال‌های MST هستند و در
پایان n-1 بار این کار را انجام می‌دهیم یال‌های
پیدا شده و ثبت می‌شوند.

الگوریتم Kruskal

یال‌های گراف را براساس وزن آن‌ها به صورت صعودی مرتب می‌کنیم و به همین ترتیب یال‌ها را به
MST اضافه می‌کنیم. اگر اضافه کردن یال یک حلقه ایجاد کند آن یال اضافه نمی‌شود.
یال min خارج MST است یکطرف (node) یال را S می‌گذاریم و بقیه node‌ها را V-S در نظر
می‌گیریم



یال 1، 2 را اضافه می‌کنیم اگر یال 3 را اضافه
کنیم حلقه می‌شود پس اضافه می‌کنیم 4 را انتخاب می‌کنیم
یا n-1 بار ادامه می‌دهیم یا می‌بینیم که اگر یال 5 را
اضافه کنیم حلقه تشکیل می‌شود پس اضافه می‌کنیم

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید

اللهم صل علی محمد وآل محمد و عجل فرجهم
WWW.MAKHFIGAH.COM دانلود رایگان سوالات دکتری،ارشد،فراگیر،پیام نور،سراسری،آزاد

Subject:

Year. Month. Date. ( )

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات دانشگاه صنعتی امیر کبیر

P4PCO

. به احترام شما دانشجویان عزیز ، پس از پرینت این جزوه هیچگونه آرم و واترمارکی مشاهده نخواهد شد
. خواهشمندیم پس از دانلود سوالات با ارسال نظرات خود،ما را در ارائه خدمات برتر به شما عزیزان یاری نمایید