فصل پنجم: برنامه‌نویسی پویا (جلسه دوم)

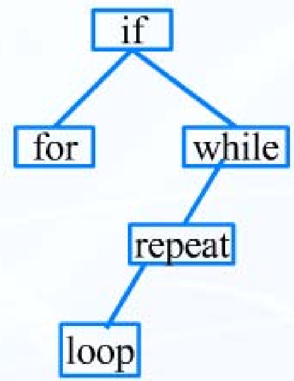
مرکز آموزش الکترونیکی
دانشگاه علم و صنعت ایران

درخت‌های دودویی جستجوی بهینه

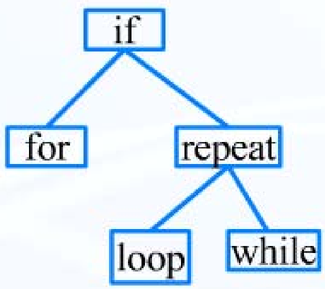
* تعریف: درخت دودویی جستجو، درختی است دودویی که اگر تهی نیست دارای شرایط زیر است:

- به ریشه کلیدی منسوب است.
- کلیدهای منسوب به زیردرخت چپ کوچکتر از کلید ریشه هستند.
- کلیدهای منسوب به زیردرخت راست بزرگتر از کلید ریشه هستند.
- زیردرخت‌های چپ و راست، هر کدام یک درخت دودویی جستجواند.

مثال



میانگین تعداد مقایسات جستجوی موفق = ۵/۱۲



میانگین تعداد مقایسات جستجوی موفق = ۵/۱۱

درخت‌های دودویی جستجوی بهینه (ادامه)

* در حالت عمومی ما با شناسه‌های با احتمالات متفاوت سروکار داریم.
* فرض کنید $a_1,a_2,a_3,\ldots,a_n$ مجموعه‌ای از n شناسه باشد که:

$$a_1<a_2<a_3<\ldots<a_n$$

* فرض کنید که p_i مساوی است با احتمال اینکه کلید خواسته شده برابر a_i باشد.

درخت‌های دودویی جستجوی بهینه (ادامه)

* فرض کنید که q_i مساوی است با احتمال اینکه کلید خواسته شده x در رابطه $a_i<x<a_{i+1}$ صادق باشد.

* فرض شده که $a_{n+1}=+\infty$ و $a_0=-\infty$

* بنابراین احتمال جستجوی موفق برابر $\sum_{i=1}^{n} p_i$ و احتمال جستجوی ناموفق برابر $\sum_{i=0}^{n} q_i$ است. بدیهی است که: $\sum_{i=1}^{n} p_i + \sum_{i=0}^{n} q_i = 1$

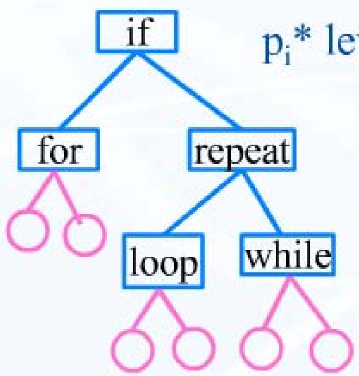
* هدف ساختن درخت جستجوی دودویی است (با استفاده از n کلید داده شده) که کمترین هزینه را داشته باشد.

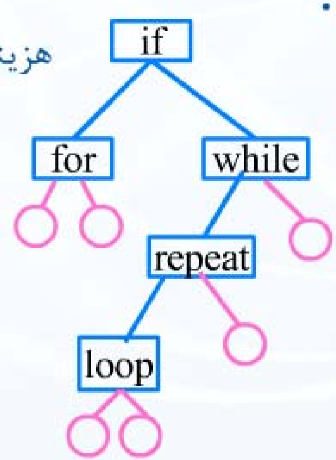
درخت‌های دودویی جستجوی بهینه (ادامه)

* به جای هریک از زیردرخت‌های تهی در درخت یک گره مجازی قرار می‌دهیم.

این گره‌ها نشان دهنده حالاتی است که عنصر مورد نظر در درخت وجود نداشته است.

هزینه گره داخلی = $p_i * level(a_i)$





درخت‌های دودویی جستجوی بهینه (ادامه)

* جستجوی ناموفق به یکی از گره‌های خارجی می‌رسد.
* شناسه‌های ناموجود در درخت جستجو را می‌توان در $n+1$ کلاس افراز نمود:

$$E_0=\{x|\ x<a_0\},\ \ E_n=\{x|\ x>a_n\},\ E_i=\{x|\ a_i<x<a_{i+1}\}$$

* اگر گره خارجی E_i در سطح $level(E_i)$ باشد، هزینه جستجوی این گره برابر $q_i*[level(E_i)-1]$ خواهد بود.

درخت‌های دودویی جستجوی بهینه (ادامه)

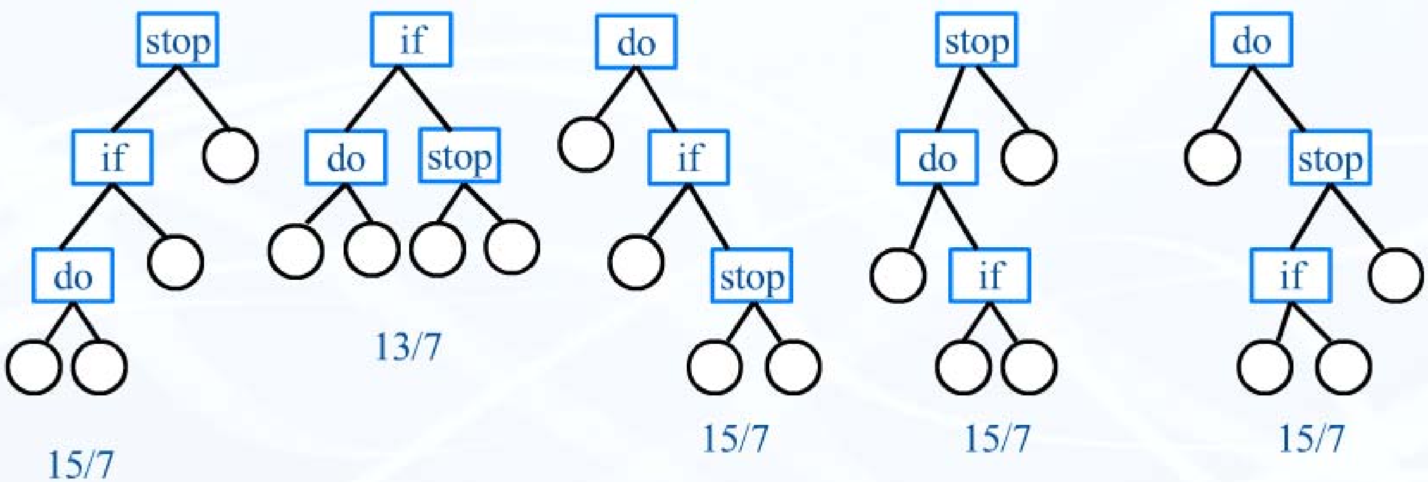
* با توجه به مطالب گفته شده، میانگین هزینه برای یک درخت دودویی جستجو مثل T برابر است با:

$$cost(T) = \sum_{i=1}^{n} p_i \times level(a_i) + \sum_{i=0}^{n} q_i \times [level(E_i) - 1)]$$

* تعریف: یک درخت جستجوی بهینه برای شناسه‌های $a_1, a_2, \ldots, a_n$ درختی است که برای آن کمیت فوق مینیمم باشد.

مثال

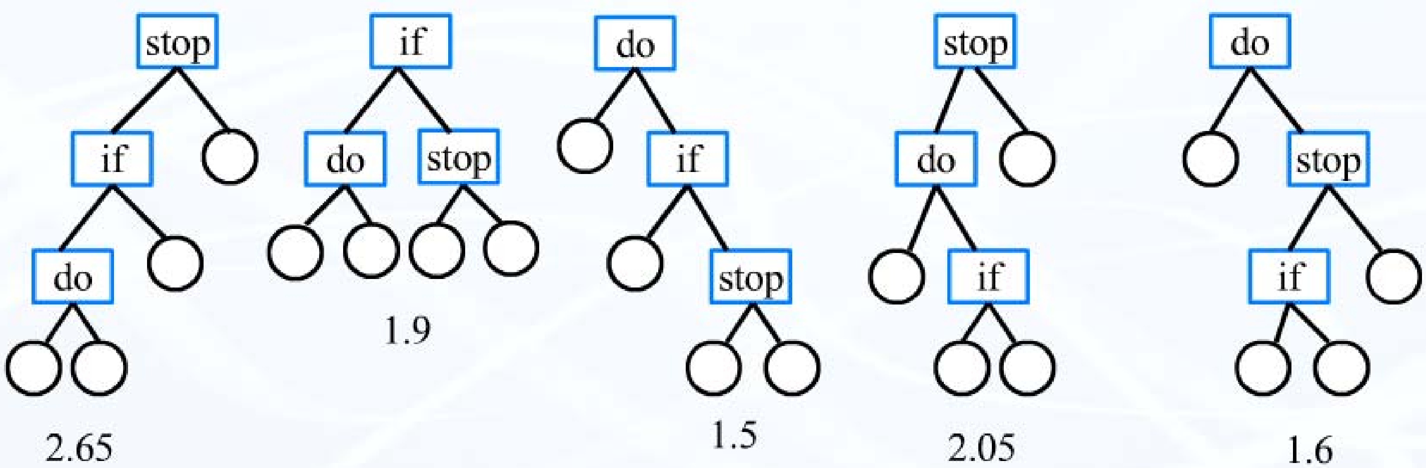
* فرض کنید که $(a_1,a_2,a_3)=(do,if,stop)$



با احتمالات مساوی $p_i=q_i=1/7$

مثال

* اگر $(q_0,q_1,q_2,q_3)=(0.15,0.1,0.05,0.05)$ و $(p_1,p_2,p_3)=(0.5,0.1,0.05)$



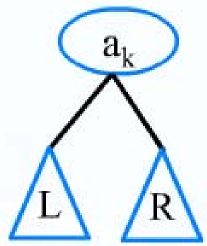
$1\times0.05+2\times0.1+3\times0.5=1.75$ $\quad$ $1\times0.05+2\times0.05+3\times0.1+3\times0.15=0.09$ $\quad$ $1.75+0.09=2.65$

الگوریتم پویا برای درخت جستجوی بهینه

* تصمیم بگیریم که کدامیک از شناسه‌های a_i را به عنوان ریشه درخت T، قرار دهیم.

* اگر a_k را انتخاب کنیم، واضح است که گره‌های داخلی a_1 تا a_{k-1} و نیز گره‌های خارجی E_0 تا E_{k-1} باید در زیردرخت سمت چپ و گره‌های داخلی a_{k+1} تا a_n همچنین گره‌های خارجی E_k تا E_n باید در زیردرخت سمت راست قرار بگیرید.

الگوریتم پویا برای درخت جستجوی بهینه



$$cost(L) = \sum_{i=1}^{k-1} p_i \times level(a_i) + \sum_{i=0}^{k-1} q_i [level(E_i) - 1]$$

$$cost(R) = \sum_{i=k+1}^{n} p_i \times level(a_i) + \sum_{i=k}^{n} q_i [level(E_i) - 1]$$

$$cost(T) = p_k + cost(L) + cost(R) + q_0 + \sum_{l=1}^{k} [p_l + q_l] + q_k + \sum_{l=k+1}^{n} [p_l + q_l]$$

الگوریتم پویا برای درخت جستجوی بهینه

* با تعریف $w(i,j) = q_i + \sum_{l=i+1}^{j} [p_l + q_l]$ هزینه میانگین برای درخت دودویی جستجو به صورت زیر بیان می‌شود:

$$cost(T)=p_k+cost(L)+cost(R)+w(0,k)+w(k,n)$$

که می‌توان آنرا به صورت زیر ساده نمود:

$$cost(T)=cost(L)+cost(R)+w(0,n)$$

* اگر $c(i,j)$ هزینه درخت جستجوی بهینه T_{ij} بر روی شناسه‌های a_i+1 تا a_j و E_i تا E_j باشد، باید $cost(R)=c(k,n)$ و $cost(L)=c(0,k-1)$.

الگوریتم پویا برای درخت جستجوی بهینه

* K باید طوری انتخاب شود که: $c(0,k-1)+c(k,n)+w(0,n)$ مینیمم شود. بنابراین:

$$c(0,n)=\min_{1<k<=n}\{c(0,k-1)+c(k,n)\}+w(0,n)$$

* در حالت کلی‌تر می‌توان فرمول فوق را به صورت زیر نوشت:

$$c(i,j)=\min_{i<k<=j}\{c(i,k-1)+c(k,j)\}+w(i,j)$$

* ضمناً $w(i,i)=q_i$ و $c(i,i)=0$

مثال

* درخت جستجوی بهینه را برای شناسه‌های (do,if,read,while) با احتمالات زیر بدست آورید.

i	P_i	q_i
0	-	2
1	3	3
2	3	1
3	1	1
4	1	1

$c(i,i)=0$

$r(i,i)=0$

$w(i,i)=q_i$

مثال (ادامه)

$j-i=1$

$w(0,1)=p_1+q_1+q_0=3+3+2=8$; $c(0,1)=\min\{c(0,0)+c(1,1)\}+w(0,1)=8$; $r(0,1)=1$

$w(1,2)=p_2+q_2+q_1=3+1+3=7$; $c(1,2)=\min\{c(1,1)+c(2,2)\}+w(1,2)=7$; $r(1,2)=2$

$w(2,3)=p_3+q_3+q_2=1+1+1=3$; $c(2,3)=\min\{c(2,2)+c(3,3)\}+w(2,3)=3$; $r(2,3)=3$

$w(3,4)=p_4+q_4+q_3=1+1+1=3$; $c(3,4)=\min\{c(3,3)+c(4,4)\}+w(3,4)=3$; $r(3,4)=4$

$j-i=2$

$w(0,2)=p_2+q_2+w(0,1)=3+1+8=12$; $c(0,2)=\min\{\underline{c(0,0)+c(1,2)}, c(0,1)+c(2,2)\}+w(0,2)=19$; $r(0,2)=1$

$w(1,3)=p_3+q_3+w(1,2)=1+1+7=9$; $c(1,3)=\min\{\underline{c(1,1)+c(2,3)}, c(1,2)+c(3,3)\}+w(1,3)=12$; $r(1,3)=2$

$w(2,4)=p_4+q_4+w(2,3)=1+1+3=5$; $c(2,4)=\min\{\underline{c(2,2)+c(3,4)}, c(2,3)+c(4,4)\}+w(2,4)=8$; $r(2,4)=3$

مثال (ادامه)

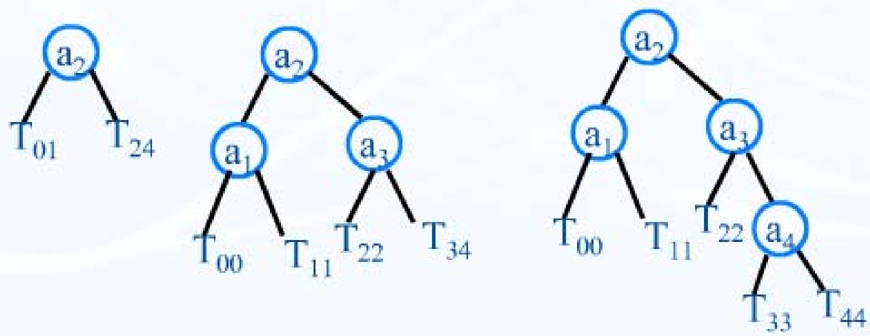
j-i=3

$w(0,3)=p_3+q_3+w(0,2)=1+1+12=14;\ c(0,3)=\min\{c(0,0)+c(1,3),\ \underline{c(0,1)+c(2,3)},$
$c(0,2)+c(3,3)\}+w(0,3)=25;\ r(0,3)=2$

$w(1,4)=p_4+q_4+w(1,3)=1+1+9=11;\ c(1,4)=\min\{\underline{c(1,1)+c(2,4)},\ c(1,2)+c(3,4),$
$c(1,3)+c(4,4)\}+w(1,4)=19;\ r(1,4)=2$

j-i=4

$w(0,4)=p_4+q_4+w(0,3)=1+1+14=16;\ c(0,4)=\min\{c(0,0)+c(1,4),\ \underline{c(0,1)+c(2,4)},$
$c(0,2)+c(3,4),\ c(0,3)+c(4,4)\}+w(1,4)=32;\ r(0,4)=2$



جواب: $c(0,4)=32$

مسأله فروشنده دوره گرد TSP

* فرض کنید که $G=(V,E,C)$ یک گراف وزن‌دار و جهت داری است که در آن $V=\{1,2,\ldots,n\}$ مجموعه رئوس و C ماتریس وزنها است. C_{ij} هزینه یال (i,j) است که به صورت زیر تعریف شده است:

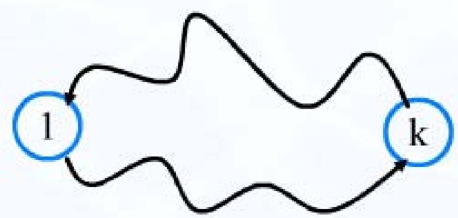
اگر $(i,j)\in E$ آنگاه $C_{ij}>0$ وگرنه $C_{ij}=\infty$

* یک دور هامیلتونی در G، مدار ساده جهت داری است که همه رئوس V را شامل باشد، هزینه یک دور هامیلتونی، برابر مجموع هزینه یالهای تشکیل‌دهنده آن است.

* مسأله TSP، یافتن یک دور هامیلتونی با هزینه مینیمم است.

مسأله فروشنده دوره گرد (ادامه)

* n انتخاب برای اولین رأس، n-1 انتخاب برای دومین رأس، n-2 انتخاب برای رأس سوم و ... .
* تعداد دورهای هامیلتونی برابر !n می‌باشد.
* اصل بهینگی:



الگوریتم پویا برای مسأله TSP

* فرض کنید که $g(i,S)$ ، طول کوتاهترین مسیری است که از رأس i شروع، از کلیه رئوس S عبور و به رأس ۱ ختم می‌شود.

بدیهی است که با توجه به این تعریف، طول یک دور هامیلتونی با هزینه مینیمم برابر $g(1,V-\{1\})$ است.

$$g(1,V-\{1\})=\min_{2<=k<=n}\{C_{1k}+g(k,S-\{k\})\}$$

$$g(i,S)=\min_{j\in S,\, i\notin S}\{C_{ij}+g(j,S-\{j\})\}$$

مثال

* برای گراف زیر یک دور هامیلتونی مینیمم بدست آورید.

$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

* حل:

$|S|=0$

$g(2,\emptyset)=C_{21}=5$ $\quad$ $g(3, \emptyset)=C_{31}=6$ $\quad$ $g(4, \emptyset)=C_{41}=8$

مثال

$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

$|S|=1$

$g(2,\{3\})=C_{23}+g(3,\emptyset)=9+6=15$ $\quad$ $g(2,\{4\})=C_{24}+g(4,\emptyset)=10+8=18$

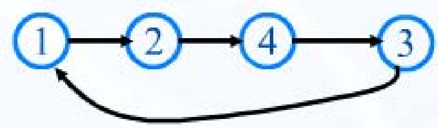
$g(3,\{2\})=18$ $\quad$ $g(3,\{4\})=20$ $\quad$ $g(4,\{2\})=13$ $\quad$ $g(4,\{3\})=15$

$|S|=2$

$g(2,\{3,4\})=\min\{C_{23}+g(3,\{4\}),\ \underline{C_{24}+g(4,\{3\})}\}=25$

$g(3,\{2,4\})=\min\{C_{32}+g(2,\{4\}),\ \underline{C_{34}+g(4,\{2\})}\}=25$

$g(4,\{2,3\})=\min\{\underline{C_{24}+g(2,\{3\})},\ C_{43}+g(3,\{2\})\}=23$



$|S|=3$

$g(1,\{2,3,4\})=\min\{\underline{C_{12}+g(2,\{3,4\})},\ C_{13}+g(3,\{2,4\}),\ C_{43}+g(4,\{2,3\})\}=35$

الگوریتم پویا برای مسأله TSP

```
void  travel (int n,const number W[][],index P[][], number& minlength)
{
     index i, j, k;
     number D[1..n][subset of V – {v1}];
     for (i = 2; i <= n; i++)
        D[i][∅ ] = W[i][1];
     for (k = 1; k <= n – 2; k++)
        for (all subsets A ⊆ V - {v1} containing k vertices)
            for (i such that i ≠ 1 and vi is not in A){
                D[i][A] = minimum(W[i][j] + D[j][A – {vj}]);
                             j : vj ∈ A
                P[i][A] = value of j that gave the minimum;
            }
      D[1][V – {v1}] = minimum (W[1][j] + D[j][V – {v1, vj}]);
                             2 ≤ j ≤ n
      P[1][V – {v1}] = value of j that gave the minimum;
      minlength = D[1][V – {v1}];
}
```

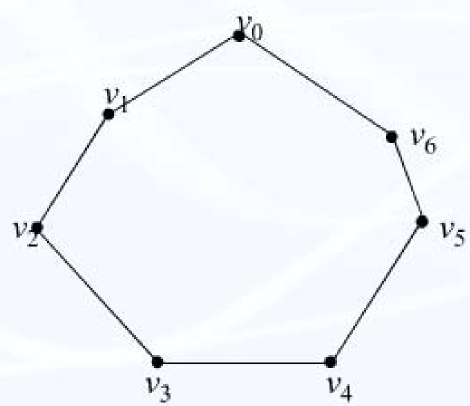
$\Rightarrow O(n^2 2^n)$

مسأله مثلث‌بندی بهینه چندضلعی‌های محدب

* یک چندضلعی محدب را با دنباله $P=<v_0, v_1, \ldots, v_{n-1}>$ از رئوس آن نمایش می دهیم. مانند شکل زیر:

* چون چندضلعی محدب است، اگر رئوس v_i و v_j مجاور نباشند، $<v_i, v_j>$ در داخل چندضلعی قرار می‌گیرد و به آن «قطر» می‌گوییم.

* این قطر چندضلعی را به چندضلعی محدب $<v_i, v_{i+1}, \ldots, v_j>$ و $<v_j, v_{j+1}, \ldots, v_i>$ تقسیم می‌کند.



مسأله مثلث‌بندی بهینه (ادامه)

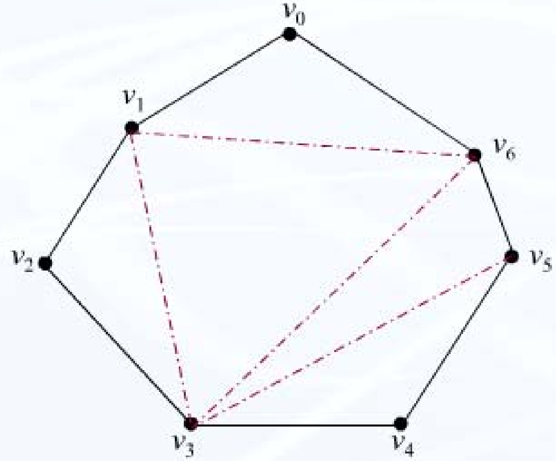
* منظور از مثلث‌بندی یک چندضلعی محدب تعیین مجموعه‌ای از قطرهای نامتقاطع است که چندضلعی را به مثلث‌های مختلفی تقسیم می‌کند.

* یک n-ضلعی همواره با $n - 3$ قطر به $n - 2$ مثلث افراز می‌شود.

* مثلث‌بندی‌ای که در آن حاصل جمع وزن‌ها کمینه باشد را مثلث بندی بهینه می‌گویند.

مسأله مثلث‌بندی بهینه (ادامه)

* یک نمونه از تابع وزن که بر روی مثلث‌ها تعریف می‌شود، چنین است:

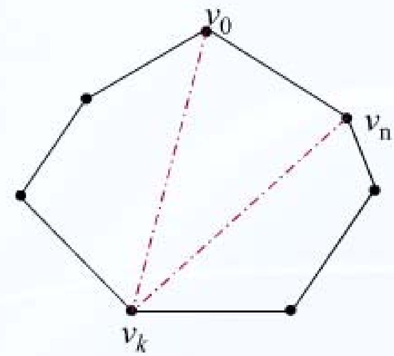


$$w(\Delta_{v_i v_j v_k}) = \overline{v_i v_j} \,.\, \overline{v_j v_k} \,.\, \overline{v_i v_k}$$

مسأله مثلث‌بندی بهینه (ادامه)

$P = <v_0, v_1, \ldots, v_n>$

$T[1][n] = ?$



* $T[i][j] =$ ارزش مثلث‌بندی بهینه برای $<v_{i-1}, v_i, \ldots, v_j>$

$$T[i,j] = \begin{cases} 0 & i = j \\ \min\limits_{i \le k \le j\ 1} (T[i][k].\ T[k.\ 1][j].\ w(\Delta(v_{i\ 1} v_j v_k))) & i < j \end{cases}$$