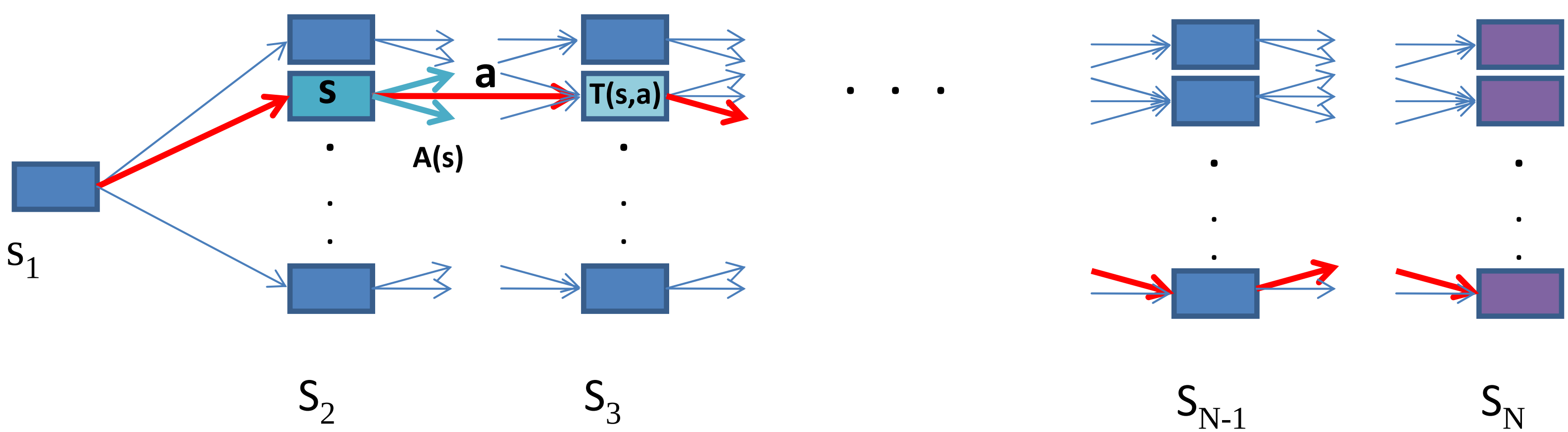
Dinamikus programozás,
mint feladatmegoldó módszer

DP feladatok/stratégiák jellemzői

1. **Részfeladatokra** bontható
   - Identikus részfeladatok
2. Érvényes az **optimalitás alapelve**
   - Nem érvényes a mohó választás alapelve
3. „**Bottom-up**” stratégia.
4. **Eltárolja** a már megoldott részfeladatok optimum-értékeit.

Optimal control (OC) / Optimal structure (OS) model

- **S**: state space (**P**: set of sub-problems)
- **A(s)**, $s \in S$: set of enabled actions (**D(p)**, $p \in P$: set of choices)
- **T(s,a)**$\subset S$, $s \in S$, $a \in A(s)$: transition function
- **r(s,a)**, $s \in S$, $a \in A(s)$: immediate returns
- Stages: 1, 2, …, N
  - $S_i \subset S$: possible states at stage i
- **f**: cost/reward – to – go/arrive function
  - $f(i,s) = opt\{g_{s,a}(r(s,a), \cup f(i+1,q)) \mid a \in A(s), q \in T(s,a)\}$, for $i=1,2,\ldots,N-1$ and $s \in S_i$
  - $f(N,s) = G_s$, $s \in S_N$



DP feladatmegoldó stratégia

„Optimal-structure” modell

1. Mi a **részfeladatok** általános (paraméteres) alakját?
2. Hol **tároljuk** az **optimum értékeket**?
3. **Rekurzív képletet** a lentről felfele építkezéshez.
4. **Leprogramozzuk** az optimális megoldás felépítését.
5. Kiolvassuk az **optimális döntéssorozatot**.

1. Feladat (optimalizálási probléma)

I. Típusú döntési fa

7.12. Virágüzlet: Legyen n különböző fajtájú virágcsokor 1-től n-ig megszámozva, és m különböző típusú váza 1-től m-ig megszámozva ($n \leqslant m$). Mind a vázák, mind a virágcsokrok *sorszámuk szerint növekvő sorrendben* vannak elhelyezve (például balról jobbra irányban) a virágüzlet kirakatában. Minden vázába egy csokrot teszünk, és ha több váza van, mint csokor, akkor egyesek üresen maradnak. Továbbá minden virágcsokornak föltétlenül a kirakatba kell kerülnie.

Egy $a[1..n][1..m]$ mátrixban adott, hogy az egyes csokrok, különböző vázákban, milyen mértékű esztétikai hatást keltenek. Pontosabban az $a[i][j]$ elem értéke annak fokmérője, hogy hogyan néz ki az i-edik virág a j-edik vázában. Egy üresen maradt váza esztétikai értéke 0.

Határozzuk meg azt a kompozíciót, amelynek összesztétikai hatása maximális.

Példa: Ha $n = 3$, $m = 5$ és az a mátrix az alábbi,

$$\begin{array}{rrrrr} 7 & 23 & -5 & -24 & 16 \\ 5 & 21 & -4 & 10 & 23 \\ -21 & 5 & -4 & -20 & 20, \end{array}$$

akkor a kihozható maximális összesztétikai hatás 53. Az optimális megoldás kódja pedig 2, 4, 5 (az 1. csokrot a 2. vázába, a 2. csokrot a 4. vázába, a 3. csokrot az 5. vázába tesszük).

1. Az első ***i*** virág optimális elhelyezése az első ***j*** vázába (i = 1..n, j = i . . m−n+i)

2. $c[0..n][0..m]$

c	0	1	2	3	4	5
0	0	0	0			
1		7	23	23		
2			28	28	33	
3				24	24	**53**

3.

$$c[0][j] = 0, \text{ ahol } j = 0, m - n$$

$$c[i][j] = c[i-1][j-1] + a[i][j], \text{ ha } i = j$$

$$c[i][j] = \max\{c[i-1][j-1] + a[i][j], c[i][j-1]\}, \text{ ha } i < j \leqslant m - n + i$$

2. Feladat (megoldások száma)

7.14. Mérleg: Adott egy különleges kétkarú mérleg. A karok elhanyagolható súlyúak és $h = 15$ hosszúak. A karokon ismert pozíciókban elhanyagolható súlyú horgok vannak, összesen n horog. A horgok pozícióit a $p[1..n]$ tömbben tároljuk ($-h \leqslant p[i] \leqslant h$, ahol $i = 1, \ldots, n$). Adott továbbá m ($1 \leqslant m \leqslant 20$) különböző értékű súly a $g[1..m]$ tömbben ($1 \leqslant g[i] \leqslant 25$, ahol $i = 1, 2, \ldots, m$). Határozzuk meg, hányféleképpen egyenlíthető ki a mérleg az összes súly felhasználásával. Egy mérleg akkor kiegyensúlyozott, ha a ráaggatott súlyok forgatónyomatékainak (súly szorozva erőkarral) összege nulla.

Példa: Ha $n = 2$ és $m = 4$, továbbá, ha a horgok a -2 és 3 pozíciókban vannak, és a súlyok rendre $3, 4, 5$ és 8 értékűek, akkor a mérleg kétféleképpen egyenlíthető ki.

1. Hányféleképpen hozható ki az első *i* súly felhasználásával a *j* mérlegérték.

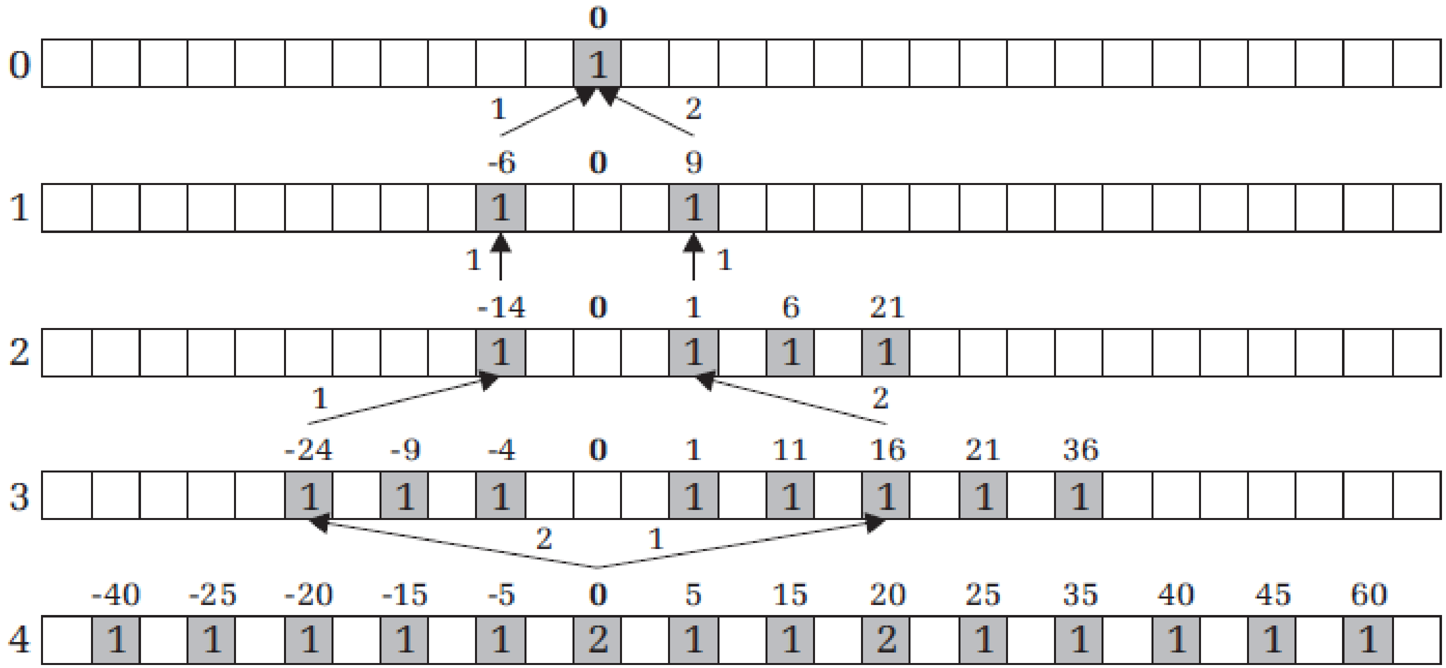
   a) *Triviális részfeladat*: nulla darab súllyal (i = 0) csakis a nullaértékű mérleg (j = 0) hozható ki.
   b) *Az eredeti feladatot:* i = m, j = 0.

2. $c[0..m][-max..max]$

3. $$c[0][0] = 1$$
   $$c[0][j] = 0, \text{ ha } -max \leqslant j \leqslant max \text{ és } j \neq 0$$

Minden $0 < i \leqslant m$ és $-max \leqslant j \leqslant max$ esetén

$$c[i][j] = \Sigma c[i-1][k] \text{ minden olyan } k = -max..max \text{ esetén, amelyre}$$
$$c[i-1][k] > 0 \text{ és létezik olyan } r = 1, 2, \ldots, n, \text{ hogy } k + g[i] \cdot p[r] = j.$$



$1, 1, 1, 2$ – az első három súlyt az első horogra, az utolsót pedig a második horogra tesszük: $(-2) \cdot 3 + (-2) \cdot 4 + (-2) \cdot 5 + 3 \cdot 8 = 0$

$2, 1, 2, 1$ – az első és harmadik súlyt a második horogra, a másodikat és a negyediket pedig az elsőre tesszük: $3{\cdot}3+(-2){\cdot}4+3{\cdot}5+(-2){\cdot}8 = 0$

```
minden i=0,m végezd
   minden j=-max,max végezd
      c[i][j]=0
   vége minden
vége minden
```

```
c[0][0]=1
minden i=0,m-1 végezd
   minden j=-max,max végezd
      ha c[i][j]≠0 akkor
         minden r=1,n végezd
            c[i+1][j+g[i+1]*p[r]] += c[i][j]
         vége minden
      vége ha
   vége minden
vége minden
```

3. Feladat (optimalizálási probléma)
II. Típusú döntési fa

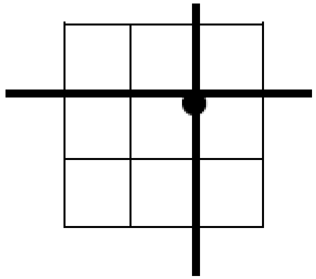
(Sapientia-ECN, 2011)

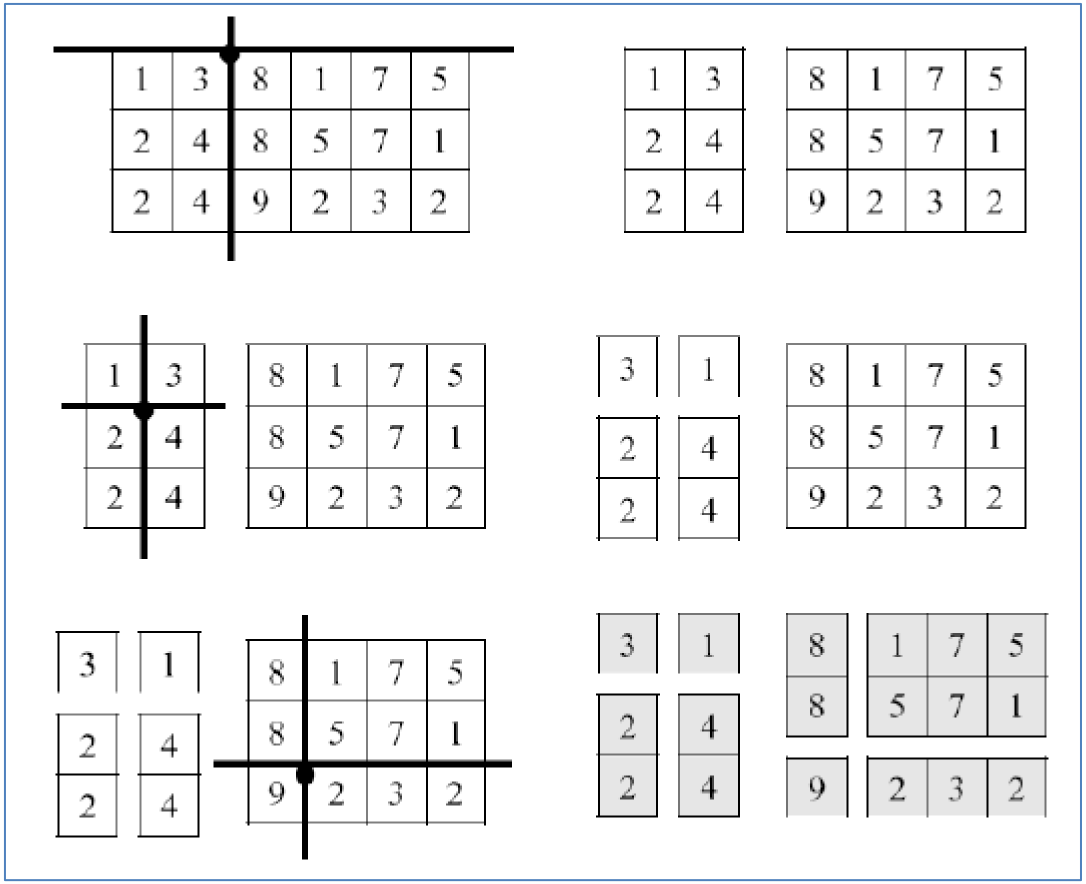
7.16. Mozaik: Egy $a[1..n][1..m]$ egész elemű tömböt úgy fogunk fel, mint egy mozaikképet. Az $a[i][j]$ tömbelem a kép (i, j) pozíciójú képeleme színkódját tárolja $(0..255)$. A képet egy golyósorozat éri, és darabokra törik az alábbi szabályok szerint:

- A golyók a képet (később a képdarabokat) a képelemek sarkainál találják el.
- Minden lövés nyomán az eltalált képdarab tovább darabolódik, ugyanis elhasad a találati ponton áthaladó vízszintes és függőleges egyenesek mentén.

Végül azt találjuk, hogy mindenik képdarab a középpontjára nézve szimmetrikus színösszetételű (a középpontra szimmetrikus pozíciójú képelemek azonos színűek).

Legkevesebb hány golyó érte a képet, és mennyi darabra esett szét? Több megoldás esetén egy olyat írjunk ki, amely esetén minimális a szimmetrikus darabok száma.





Sample Input: F.IN

```
2
3 6
1 3 8 1 7 5
2 4 8 5 7 1
2 4 9 2 3 2
2 2
1 2
3 4
```

Sample Output: Standard output

```
3 8
1 1 1 1
1 2 1 2
2 1 3 1
2 2 3 2
1 3 2 3
3 3 3 3
1 4 2 6
3 4 3 6
1 4
```

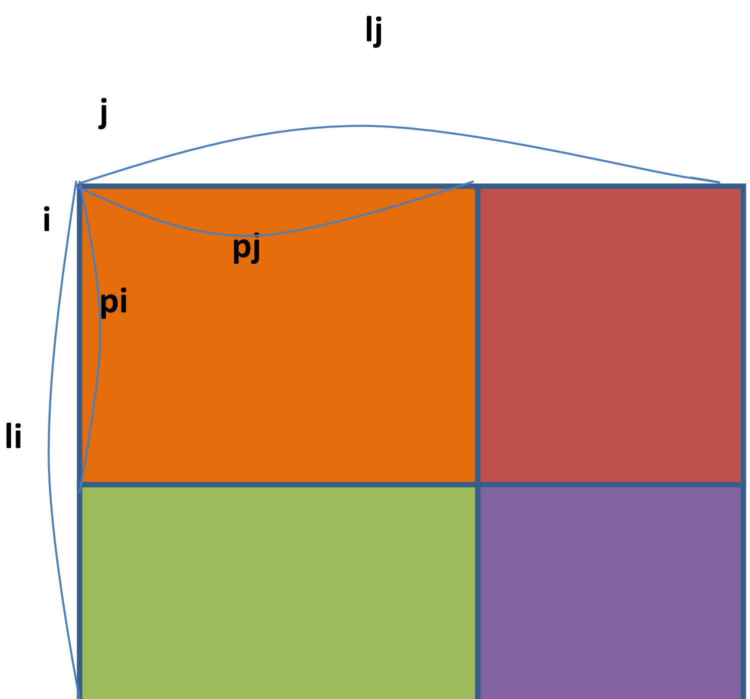
1. Az (*i,j*) sarkú, *li* és *lj* oldalhosszú téglalap optimális feldarabolása.
   a) *Triviális,* ha az (i,j,li,lj) téglalap szimmetrikus.
   b) *Eredeti feladat*: (1,1,n,m)

2. c[1..32][1..32][1..32][1..32]
   - c[i][j][li][lj] cella tárolja az *általános részfeladat* optimum értékét.
   - *Eredeti feladat* optimum értéke: c[1][1][n][m].

3. **c[i][j][li][lj] = min** {**c[i][j][pi][pj]+**
   c[i+pi][j][li-pi][pj]+
   c[i][j+pj][pi][lj-pj]+
   c[i+pi][j+pj][li-pi][lj-pj]}

pi=0,li-1

pj=0,lj-1



```
for(li=1;li<=n;li++) //összes téglalap méretük szerint növekvő sorrendben
 for(lj=1;lj<=m;lj++)
  for(i=0;i<=n-li;i++)
   for(j=0;j<=m-lj;j++)
    if (simetric(i,j,li,lj))
     {c[i][j][li][lj]=0;              b[i][j][li][lj].pi=-1; b[i][j][li][lj].pj=-1;}
    else
    {
     min=n*m;
     for(pi=0;pi<li;pi++) //kurrens téglalap összes potenciális golyó-pozíciója
      for(pj=0;pj<lj;pj++)
      {
       if(!(pi==0 && pj==0))
       {
        x=c[i][j][pi][pj]+c[i+pi][j][li-pi][pj]+c[i][j+pj][pi][lj-pj]+c[i+pi][j+pj][li-pi][lj-pj];
        if (x<min) {min=x;     mpi=pi;mpj=pj;}
       }
      }
     a[i][j][li][lj]=min+1;        b[i][j][li][lj].pi=mpi; b[i][j][li][lj].pj=mpj;
    }
```

4. Feladat (utazó-kereskedő probléma)
„Optimal-control” modell

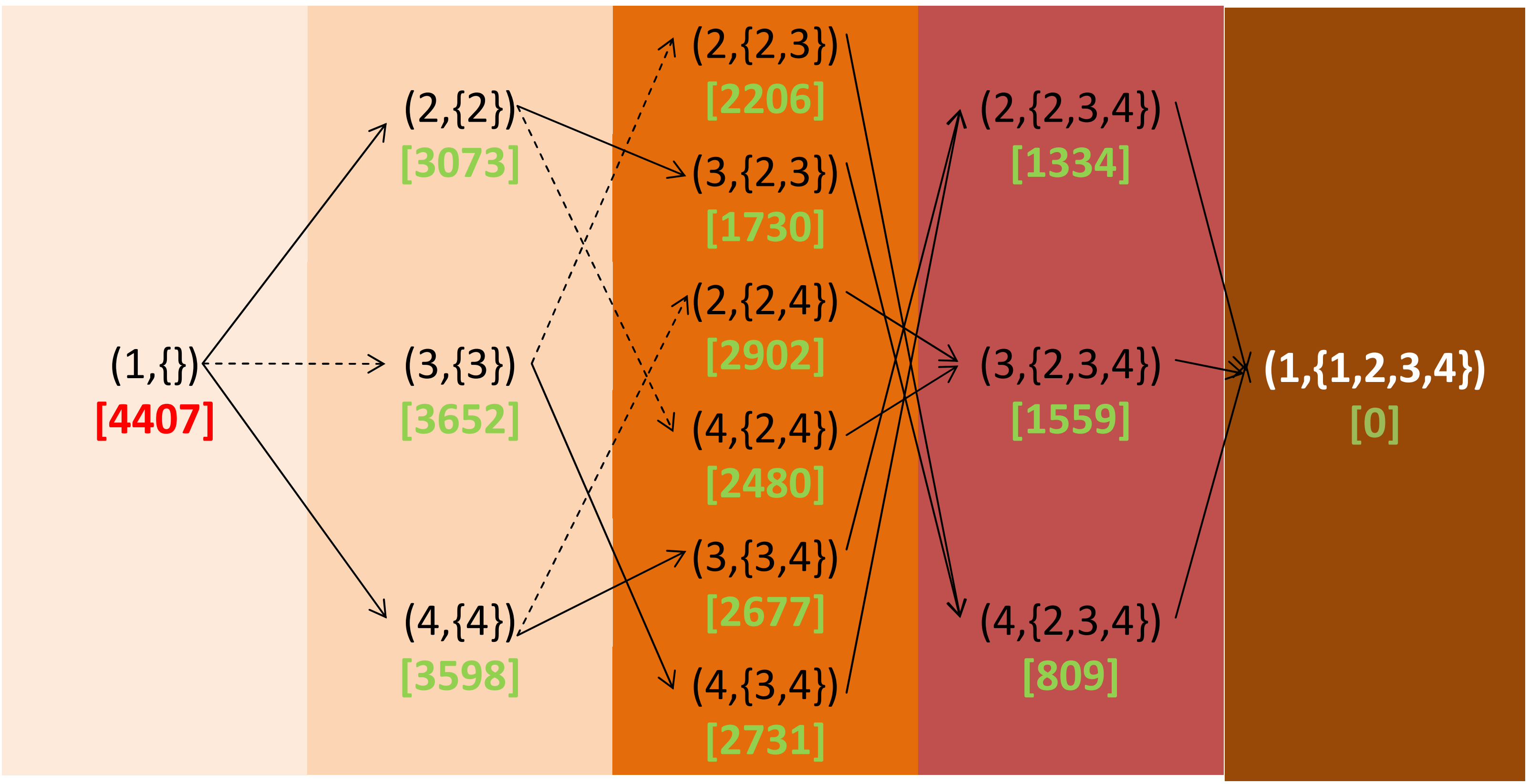
	New York	Miami	Dallas	Chicago
1. New York	—	1334	1559	809
2. Miami	1334	—	1343	1397
3. Dallas	1559	1343	—	921
4. Chicago	809	1397	921	—

$$f_t(i, S) = \min_{j \neq 1 \text{ and } j \notin S} \{c_{ij} + f_{t+1}(j, S \cup j)\}.$$

	{}	{2}	{3}	{4}	{2,3}	{2,4}	{3,4}	{2,3,4}
1	4407							
2		3073			2206	2902		1334
3			3652		1730		2677	1559
4				3598		2480	2731	809

$$\sum_{k=1}^{n-1}(k * {}_{n-1}C^{k}) + 2 \ll (n-1)!$$

4. Feladat (utazó-kereskedő probléma)



Dimenziók átka (curse of dimensionality)

20 város (10. „stage”): állapotok_száma > 1 millió

30 város (15. „stage”): állapotok_száma > 1 billió

100 város (50. „stage”): állapotok_száma > 5.000.000.000.000.000.000.000.000.000.000

Referenciák

Kátai Zoltán, **Algoritmusok felülnézetből**, Scientia Kiadó, Kolozsvár, 2007.

Kátai Zoltán, **Gráfelméleti algoritmusok**, Scientia Kiadó, Kolozsvár, 2008.

http://www.ms.sapientia.ro/~katai_zoltan/