



Bűvös huszonhat

KÉSZÍTETTE: OSZKOCSIL KRISZTIÁN

Probléma bemutatása

1-től 12-ig minden egész számot egyszer kell egy kereszt alakú táblázatba úgy elhelyeznem, hogy a középső 2 sor, a középső 2 oszlop, és a hasonló színekkel jelölt mezők összege a bűvös 26-ot adja.

$$a1+a2+a3+a4=26$$

$$b1+b2+b3+b4=26$$

$$c1+c2+c3+c4=26$$

$$b1+c1+c2+b3=26 \rightarrow$$

$$b2+c3+c4+b4=26 \rightarrow$$

	a1	a2	
b1	c1	c2	b3
b2	c3	c4	b4
	a3	a4	

$$a1+c1+c3+a3=26$$

$$a2+c2+c4+a4=26$$



Állapottér

- ▶ A mezők üresen állnak, amíg nem helyezünk bele számokat.

Az állapotokban minden mező felvesz egy

$\{-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ értékeket.

-1: nem változhat

Nulla értéket az üres cella vesz fel.

- ▶
$$A = \left\{ \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \in H^{4 \times 4} \mid i=1, j=4 \wedge a_{i,i}, a_{i,j}, a_{j,i}, a_{j,j} = -1 \wedge \forall k \forall l (i \neq k \wedge j \neq l \supset a_{kl} > -1) \right\}$$

Kezdőállapot

-1	0	0	-1
0	0	0	0
0	0	0	0
-1	0	0	-1

Célállapotok halmaza:

$$C = \left\{ \left[\begin{array}{cccc} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{array} \right] \in A \mid \begin{array}{l} \sum_{i=1}^4 a_{2,i} = 26 \wedge \sum_{i=1}^4 a_{3,i} = 26 \wedge \\ \sum_{i=1}^4 a_{i,2} = 26 \wedge \sum_{i=1}^4 a_{i,3} = 26 \wedge \\ \sum_{i=1}^4 \sum_{j=1}^4 a_{i,j} = (78-4) \wedge \\ a_{1,2} + a_{1,3} + a_{4,2} + a_{4,3} = 26 \wedge \\ a_{2,1} + a_{3,1} + a_{2,4} + a_{3,4} = 26 \wedge \\ a_{2,2} + a_{2,3} + a_{3,2} + a_{3,3} = 26 \end{array} \right\}$$

Operátorok halmaza

- Egy operátor, három komponens:

első két helyen a mező pozíciója, amelyre helyezzük, a harmadik részét, magát az értéket:

-1	0	0	-1
0	0	0	0
0	0	0	0
-1	0	0	-1

$\circ_{1,3,5}$



-1	0	5	-1
0	0	0	0
0	0	0	0
-1	0	0	-1

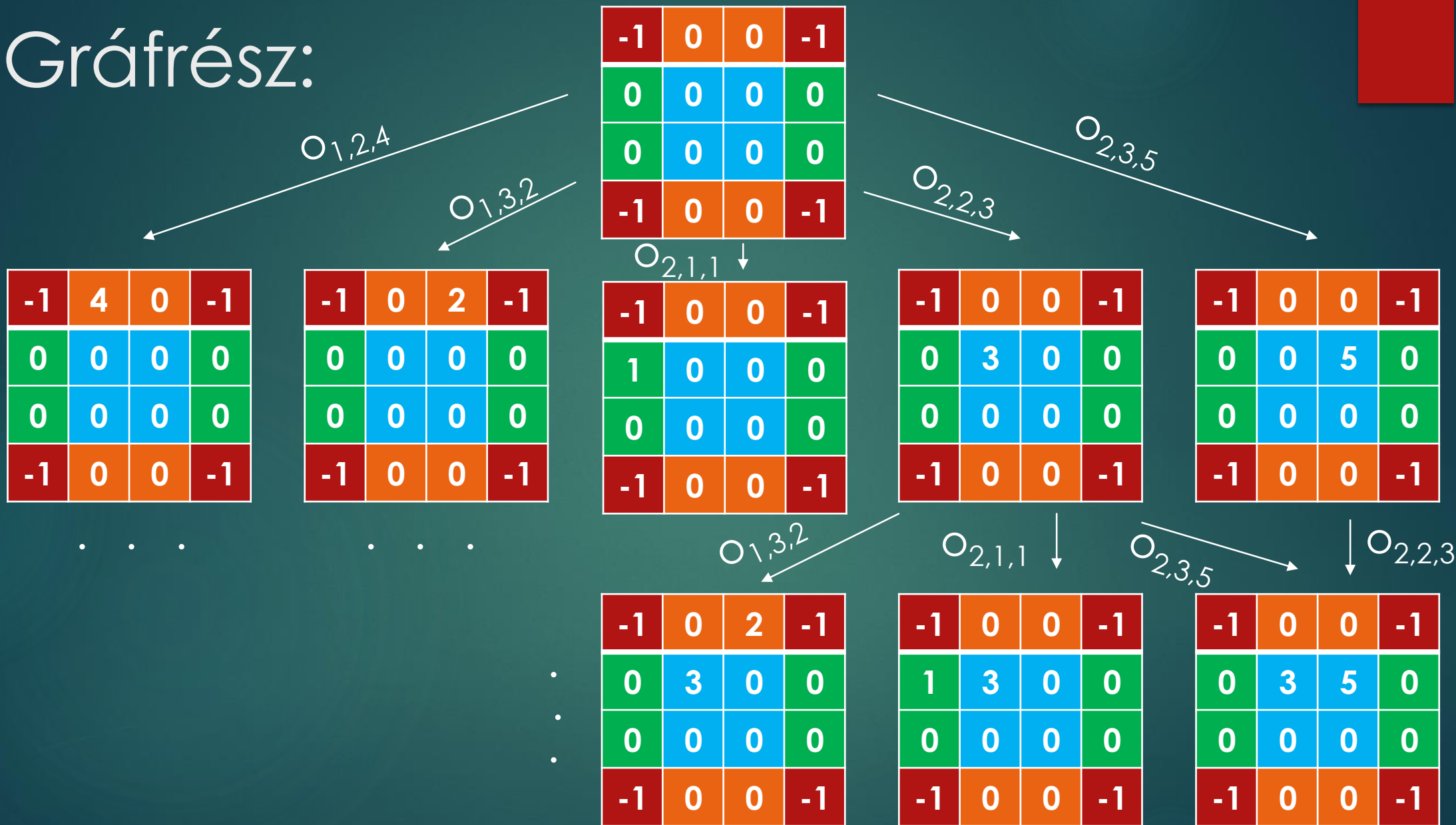
Alkalmazási előfeltétel és hatásdefiníció

► $O = \{o_{i,j,n} \mid i \in \{1,2,3,4\} \wedge j \in \{1,2,3,4\} \wedge n \in \{1,2,\dots,12\}\}$

$$\text{dom}(o_{i,j,n}) = \left\{ \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \in A \mid \exists i \exists j (a_{i,j} = 0 \wedge \forall k \forall l (a_{k,l} \neq n \wedge k \neq i \wedge l \neq j)) \right\}$$

$$o_{i,j,n} \left\{ \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \right\} = \left\{ \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} \end{bmatrix} \right\} \begin{cases} n & , \text{ha } a_{j,k}=0 \\ b_{k,l} & \text{egyébként} \end{cases}$$

Gráfrész:

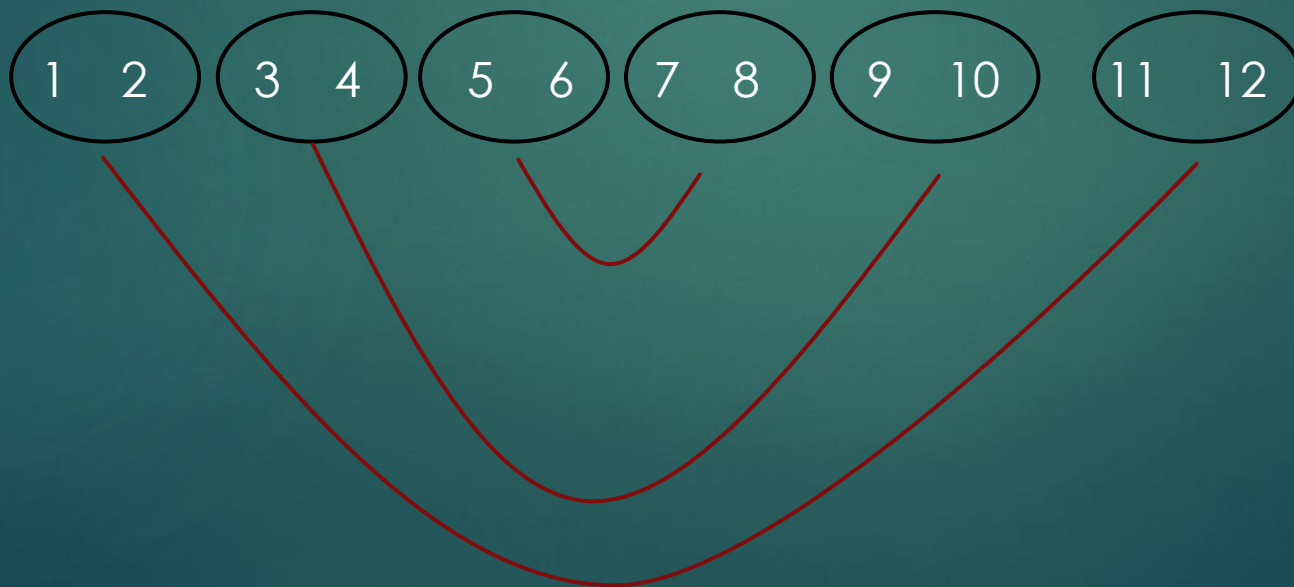


Keresők, optimalizálás

12 lépésből meg lehet oldani

Backtrack alkalmazása

Optimalizálás:



Implementálás

Kezdőállapot:

```
int[][] table = {{-1,0,0,-1},{0,0,0,0},{0,0,0,0},{-1,0,0,-1}};
```

Szabályok:

```
if(sum == 74)
{
    //table[1][0]+table[1][1] + table[1][2] + table[1][3] =26;
    if(table[1][1] + table [1][2] + table[2][1] + table[2][2] == 26 &&
        table[0][1]+table[0][2] + table[3][1]+table[3][2] == 26 &&
        table[1][0] + table[2][0] + table[2][3] + table[1][3] == 26 &&
        table[0][1] + table[1][1] + table[2][1] + table[3][1] == 26 &&
        table[0][2] + table[1][2] + table[2][2] + table[3][2] == 26 &&
        table[1][0]+table[1][1] + table[1][2] + table[1][3] ==26)
    {
        return true;
    }
}
```

Implementálás:

Backtrack használata

```
private static List<Operator> result(Node node)
{
    List<Operator> result = new ArrayList<>();
    while (node.getParent() != null)
    {
        result.add(node.getCreator());
        node = node.getParent();
    }
    return result;
}
```

```
protected Object clone(){
    Operator temp = new Operator();
    temp.setCol(col);
    temp.setRow(row);
    temp.setValue(value);
    return temp;
}
```

```
public Node(Condition condition, Node parent, List<Operator> operators, Operator creator)
{
    this.condition = (Condition) condition.clone();
    this.operators.addAll(operators);
    this.parent = parent;
    if (creator != null)
    {
        this.creator = (Operator) creator.clone();
    }
}
```

Implementálás:

Eredmény:

```
New node was created
-1 1 2 -1
3 5 8 10
6 9 4 7
-1 11 12 -1
```

Megoldás:

```
-1 1 2 -1
3 5 8 10
6 9 4 7
-1 11 12 -1
[row:3 colomn: 2 value: 12
, row:3 colomn: 1 value: 11
, row:2 colomn: 3 value: 7
, row:2 colomn: 2 value: 4
, row:2 colomn: 1 value: 9
, row:2 colomn: 0 value: 6
, row:1 colomn: 3 value: 10
, row:1 colomn: 2 value: 8
, row:1 colomn: 1 value: 5
, row:1 colomn: 0 value: 3
, row:0 colomn: 2 value: 2
, row:0 colomn: 1 value: 1
```



Köszönöm a figyelmet!

Jöhetnek a kérdések!

