SZÁLLÍTÁSI FELADAT

c_{ij}

1 x_{11}	3 x_{12}	2 x_{13}	5 x_{14}	4 x_{15}	$15 = a_1$
3 x_{21}	8 x_{22}	2 x_{23}	1 x_{24}	3 x_{25}	$17 = a_2$
1 x_{31}	4 x_{32}	2 x_{33}	5 x_{34}	1 x_{35}	$23 = a_3$
3 x_{41}	5 x_{42}	1 x_{43}	8 x_{44}	7 x_{45}	$12 = a_4$
$b_1 = 10$	$b_2 = 11$	$b_3 = 13$	$b_4 = 18$	$b_5 = 15$	$\Sigma = 67$

raktárak készlete

bolt igénye

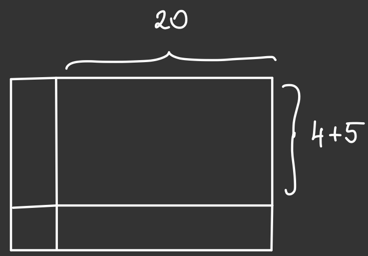
1. Egyféle árut szállítunk
2. Minden sor megfelel egy raktárnak
3. Minden oszlop megfelel egy boltnak
4. A raktárkészlet megegyezik a bolt összes igényével

FELADAT: Döntsük el, hogy melyik raktárból melyik boltba mennyit szállítsunk úgy, hogy:
- a raktárkészletet nem léphetjük túl
- minden bolt kapja meg, amit igényelt
- az összes költség minimális legyen

1.) A feladat modellje:

$$
\begin{cases}
\sum_j x_{ij} = a_i & (\forall i) \\
\sum_i x_{ij} = b_{ij} & (\forall j) \\
\underline{x} \geq \underline{0} & \\
\sum_{ij} c_{ij} x_{ij} \to \min &
\end{cases}
$$

Szimplex tábla: (2. fázisban)



$$
\begin{cases}
A\underline{x} = b \\
\underline{x} \geq \underline{0} \\
z = \underline{c}^T \underline{x} \to \min
\end{cases}
$$

<u>Állítás:</u> Ez egy LP. (lineáris programozási feladat)

<u>Állítás:</u> Az együttható mátrix rangja: $n+m-1$

2.) Kétfázisú módszerrel oldjuk meg a feladatot:

- 1. fázis: megengedett bázismegoldás előállítása
- 2. fázis: báziscserékkel optimális megoldás megadása

Az első fázis algoritmusa:

- É-NY-i módszer ← most ezt használjuk
- Sor-minimum módszer
- Tábla-minimum módszer
- Vogel-Korda

1. fázis

1 (10)	3 (5)	2	5	4	~~15~~ ~~5~~
3	8 (6)	2 (11)	1	3	~~17~~ ~~11~~
1	4	2 (2)	5 (18)	1 (3)	~~23~~ ~~21~~ ~~3~~
3	5	1	8	7 (12)	~~12~~
~~10~~	~~11~~ ~~6~~	~~13~~ ~~2~~	~~18~~	~~15~~ ~~12~~	$\Sigma = 67$

2. fázis:

1 (10)	3 (5)	2	5	4
3	8 (6)	2 (11)	1 (x)	3
1	4	2 (2)	5 (18)	1 (3)
3	5	1	8	7 (12)

$1 - 5 + 2 - 2 = -4 < 0$

jó csere

1 (10)	3 (5)	2 (x)	5	4
3	8 (6)	2	1 (11)	3
1	4	2 (13)	5 (7)	1 (3)
3	5	1	8	7 (12)

$2-2+5-1+8-3=9>0$

nem jó csere

Báziscella: (kő) ahol szállítunk valamit
(Az algoritmus neve: STEPPING STONE)

Kör:
- nem kőről indulunk
- felváltva lépünk függőlegesen és vízszintesen
- csak kövekre szabad lépni
- visszaérünk oda, ahonnan indultunk

Állítás: Tetszőleges nem kőről indulva pontosan 1 kör van.

Állítás: Ha nincs javító csere, akkor optimális a megoldás.

Segéd algoritmus: segít megtalálni a javítócserét.
- minden sorhoz: u_i
- minden oszlophoz: v_j

$u_i + v_j = c_{ij} \quad \forall (i,j) \in B$

Az egyenletrendszernek mindig van megoldása és a szabadságfoka 1.

	v_1	v_2	v_3	v_4	v_5
u_1	1 10	3 5	2	5	4
u_2	3	8 6	2	1 11	3
u_3	1	4	2 13	5 7	1 3
u_4	3	5	1	8	7 12

	1	3	−7	−4	−8
0	1 10	3 5	2 +	5 +	4 +
5	3 18 −3	8 6	2 +	1 11	3 +
9	1 54 −9	4 48 −8	2 13	5 7	1 3
15	3 78 −13	5 78 −13	1 84 −7	8 21 −3	7 12

$2-(-7)-0=9$ +
$5-(-4)-0=9$ +
$4-(-8)-0=12$ +
$3-1-5=-3$ −
$2-(-7)-5=4$ +
$3-(-8)-5=7$ +
$1-1-9=-9$ −
$4-3-9=-8$ −

$3-1-15=-13$ −
$5-3-15=-13$ −
$8-(-4)-15=-3$ −
$8-(-4)-15=-3$ −

$3\cdot 6=18$
$6\cdot 9=54$
$8\cdot 6=48$
$13\cdot 6=78$
$7\cdot 12=84$
$3\cdot 7=21$

Állítás: a redukált költség számolása: $c_{ij} - (u_i + v_j)$

Vogel-Korda:

1 / 4	3 / 11	2	5	4	~~15~~ ~~1~~ ①	← 1 2
3	8	2	1 / 17	3	~~17~~ ~~1~~	← 1 2
1 / 6	4	2 / 1	5 / 1	1 / 15	23 ~~0~~ 1	← 1 1
3	5	1 / 12	8	7	~~12~~ ~~2~~ ②	← 3 1
~~10~~ 6	~~11~~	~~13~~ 1	~~18~~ 1	~~15~~	$\sum = 67$	
1 1 → 0	3 4 → 1	1 2 → 1	1 5 → 4	1 3 → 2		
0 0 0	1 1 ①	1 1 0	0 ④ 0	2 ③		

① Felírjuk soronként és oszloponként a 2 legkisebb költség különbségét

② Kiválasztjuk a legnagyobbat

③ A legolcsóbb cellát megkeressük

④ Elviszünk annyit, amennyit lehet

Nem képes mindig optimális megoldást találni.

	1	3	2	5	1
0	1 4	3 11	2 0	5 0	4 +
−4	3 +	8 +	2 +	1 17	3 +
0	1 6	4 +	2 1	5 1	1 15
−1	3 +	5 +	1 12	8 +	7 +

Ez bázis megoldás és optimális megoldás is egyben.

$$z = 1\cdot 4 + 3\cdot 11 + 1\cdot 17 + 2\cdot 1 + 5\cdot 1 + 1\cdot 15 + 1\cdot 12 = \underline{\underline{88}}$$