



# Projektütemezés

Virtuális vállalat  
2013-2014 1. félév 5. gyakorlat  
Dr. Kulcsár Gyula

# Projektütemezési feladat megoldása

# Projektütemezés

- Projekt:
  - Egy nagy, összetett, általában egyedi igény alapján előállítandó termék vagy nyújtandó szolgáltatás előállítására/teljesítésére irányú törekvés, amely általában nagyszámú komponens feladat/aktivitás végrehajtását igényli.
- Projektütemezés:
  - Projekt(ek) időbeli végrehajtásának megtervezése úgy, hogy a megfogalmazott célok teljesüljenek figyelembe véve az előírt korlátozásokat.

# Projektütemezés jellemzői

- Cél:
  - egy vagy többcélú optimalizálás,
  - amelyben sokféle szempont szerepelhet (pl. minőség, idő, költség, felhasználói elégedettség stb.).
- Feladatok/aktivitások hálózata alakul ki (pl. megelőzési relációk alapján).
- Korlátozottan/korlátlanul rendelkezésre álló erőforrásokat kell figyelembe venni.

# Projekt példák

- Termelés
- Tervezés
- Kutatás/fejlesztés
- Menedzsment
- Építés
- Karbantartás, fenntartás
- Implementálás, telepítés
- stb.

# Projektütemezés alapjai

- Projekt/projektek reprezentálása (precedencia gráfok)
- Modellek és megoldási módszerek
  - Kritikus útvonal módszer (egyszerű) (Critical Path Method, CPM)
  - Erőforrás korlátozott projektütemezés (bonyolult) (Resource-Constrained Project Scheduling, RCPS)
    - Prioritás/szabályalapú megoldási módszerek
    - Tudás-intenzív megoldási módszerek
  - Kiterjesztett modellek és módszerek (összetett)

# Projekt reprezentálása precedencia gráffal

## **„job on node” ábrázolás**

Csomópont: feladat

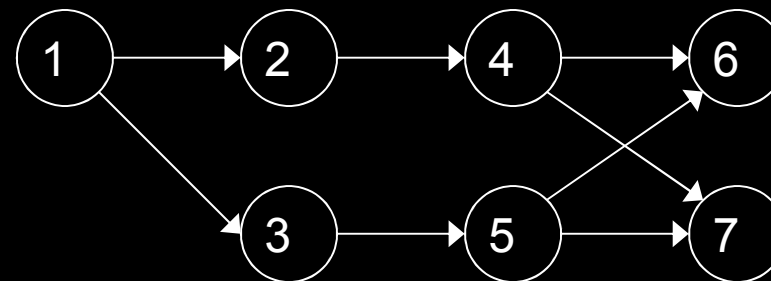
A csomópontok számozottak.

Írányított él: kötelező  
sorrendiség

Nincs irányított körút.

Nincs redundáns él.

Feladat	Végrehajtási idő [időegység]	Megelőző feladat(ok)
1	2	-
2	3	1
3	1	1
4	4	2
5	2	3
6	1	4, 5
7	3	4, 5



# Projektütemezési feladat erőforráskorlátok nélkül

- Feltételezzük, hogy:
  - korlátlan erőforrások állnak rendelkezésre párhuzamosan.
  - adott  $n$  feladat megelőzési relációkkal.
  - minden egyes feladat  $p_j$  végrehajtási idejét ismertjük.
- Az ütemezés célja:
  - a projekt befejezési időpontjának (makespan) minimalizálása.



# Projektütemezési feladat erőforráskorlátok nélkül

- A  $j$  feladat:
  - végrehajtási ideje:  $p_j$
  - legkorábbi lehetséges kezdési időpontja:  $S'_j$
  - legkorábbi lehetséges befejezési időpontja:  $C'_j$
  - legkésőbbi megengedett befejezési időpontja:  $C''_j$
  - időtartaléka:  $slack_j = C''_j - p_j - S'_j$
- Kritikus feladat: nincs tartaléka  $slack_j = 0$
- Kritikus útvonal: kritikus feladatok láncolata.<sup>9</sup>

# Kritikus útvonal módszer (Critical Path Method, CPM)

- A CPM módszer két algoritmusból áll:
  - „Forward procedure”
  - „Backward procedure”

# Kritikus útvonal módszer (Critical Path Method, CPM)

- Előre haladó eljárás  
(Forward procedure):
  - Kezdeti időpontból indul, a precedencia gráfon végighaladva az irányított élek mentén kiszámítja minden feladat esetében a legkorábbi megengedett indítási és befejezési időpontot.
  - Az utolsónak elkészülő feladat adja meg a projekt befejezési időpontját.

# Előre haladó eljárás (Forward procedure)

## 1. lépés:

Legyen  $t = t_s$  (pl.  $t_s = 0$  az indítás referencia időpontja).  
A megelőző feladattal nem rendelkező minden egyes  $j$  feladat esetében legyen  $S_j' = t$  és  $C_j' = t + p_j$ .

## 2. lépés:

A megelőző feladattal rendelkező minden egyes  $j$  feladat esetében legyen induktív módon:

$$S_j' = \max_{all\ k \rightarrow j} C_k' \quad \text{és} \quad C_j' = S_j' + p_j.$$

## 3. lépés:

A legkorábbi projekt-befejezési időpont:

$$C_{max} = \max \{C_1', C_2', \dots, C_n'\}$$

# Kritikus útvonal módszer (Critical Path Method, CPM)

- Visszafelé haladó eljárás  
(Backward procedure):
  - A projekt befejezési időpontjából indul, a precedencia gráfon az irányított élek mentén visszafelé haladva kiszámítja minden feladat esetében a legkésőbbi megengedett befejezési és indítási időpontot tekintettel arra, hogy a projektbefejezési határidő még tartható legyen.

# Visszafelé haladó eljárás (Backward procedure)

## 1. lépés:

Legyen  $t = C_{max}$

A rákövetkező feladattal nem rendelkező minden egyes  $j$  feladat esetében legyen

$$C_j'' = C_{max} \text{ és } S_j'' = C_{max} - p_j.$$

## 2. lépés:

A rákövetkező feladattal rendelkező minden egyes  $j$  feladat esetében legyen

$$C_j'' = \min_{k \rightarrow all j} S_k'' \text{ és } S_j'' = C_j'' - p_j.$$

## 3. lépés:

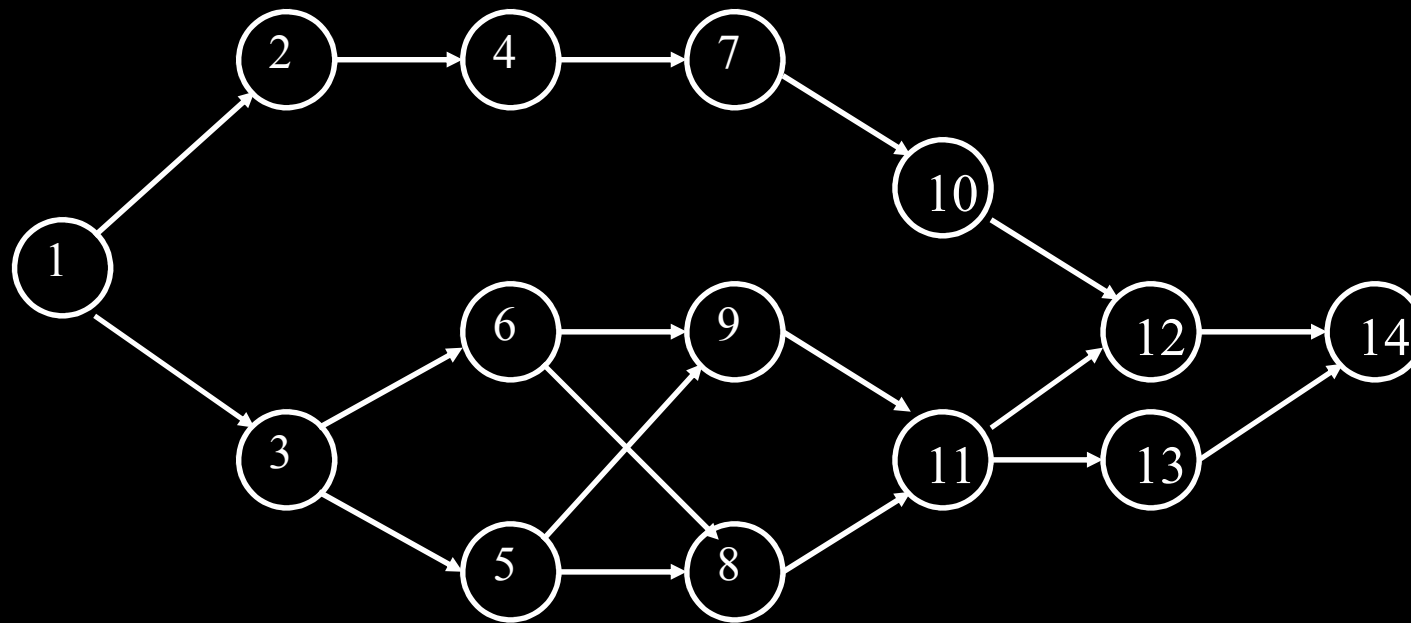
Ellenőrizzük, hogy  $t_s = \min\{S_1'', \dots, S_n''\}.$

# Magyarázat

- A *forward procedure* megadja az  $S_j'$  megengedett legkorábbi indítási időpontját minden feladatnak.
- A *backward procedure* megadja az  $S_j''$  megengedett legkésőbbi indítási időpontját minden feladatnak.
- Ha ezek azonosak akkor a feladat kritikus.
- Ha ezek különbözőek akkor a feladatnak van tartaléka (**slack**).
- Kritikus útvonal (**critical path**):  
kritikus feladatok láncolata, amely a  $t_s$  kezdési időponttól a  $C_{max}$  befejezési időpontig vezet.
- Kritikus útvonalból egyszerre több is lehet, ezek akár részben fedhetik is egymást.

# CPM példa 1

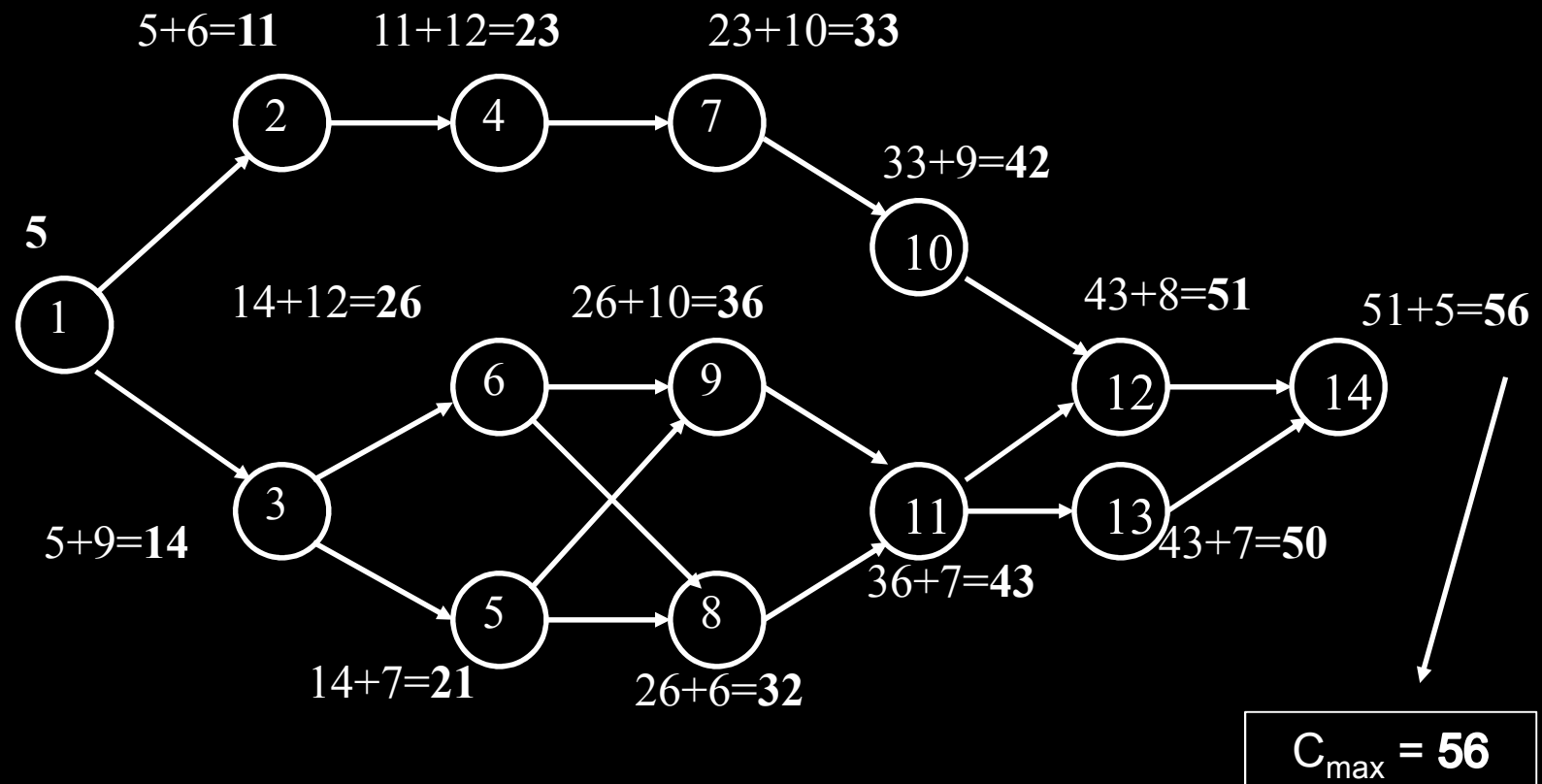
$j$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$p_j$	5	6	9	12	7	12	10	6	10	9	7	8	7	5





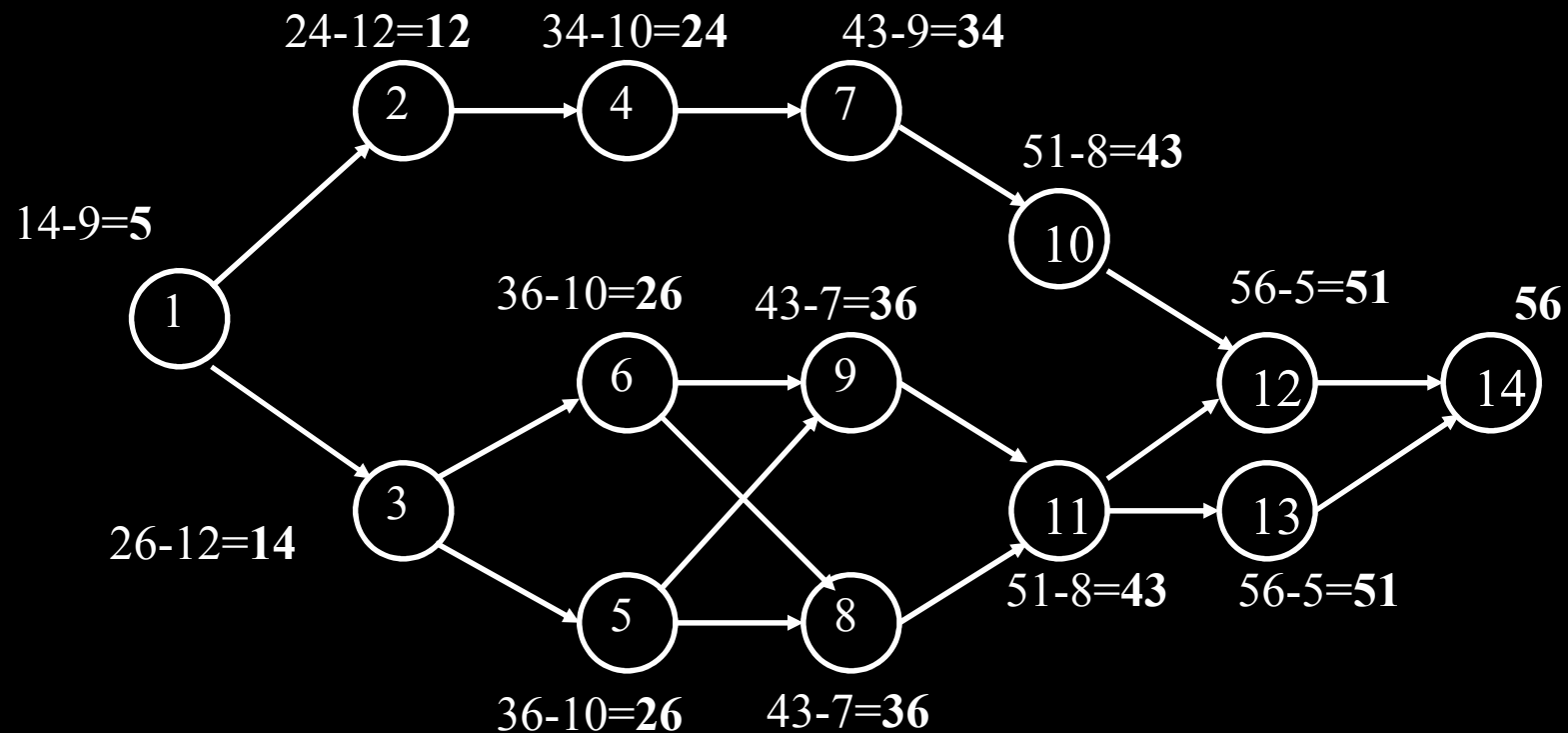
# Forward Procedure példa 1

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p <sub>j</sub>	5	6	9	12	7	12	10	6	10	9	7	8	7	5

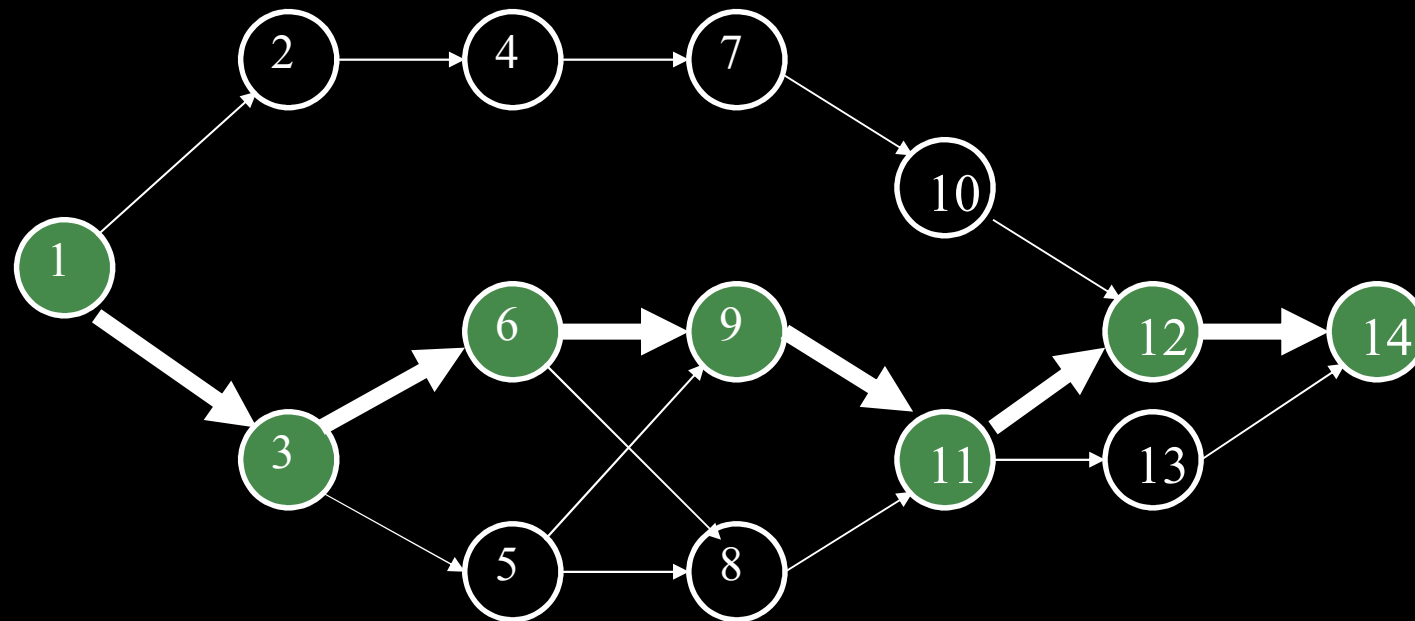


# Backward Procedure példa 1

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p <sub>j</sub>	5	6	9	12	7	12	10	6	10	9	7	8	7	5



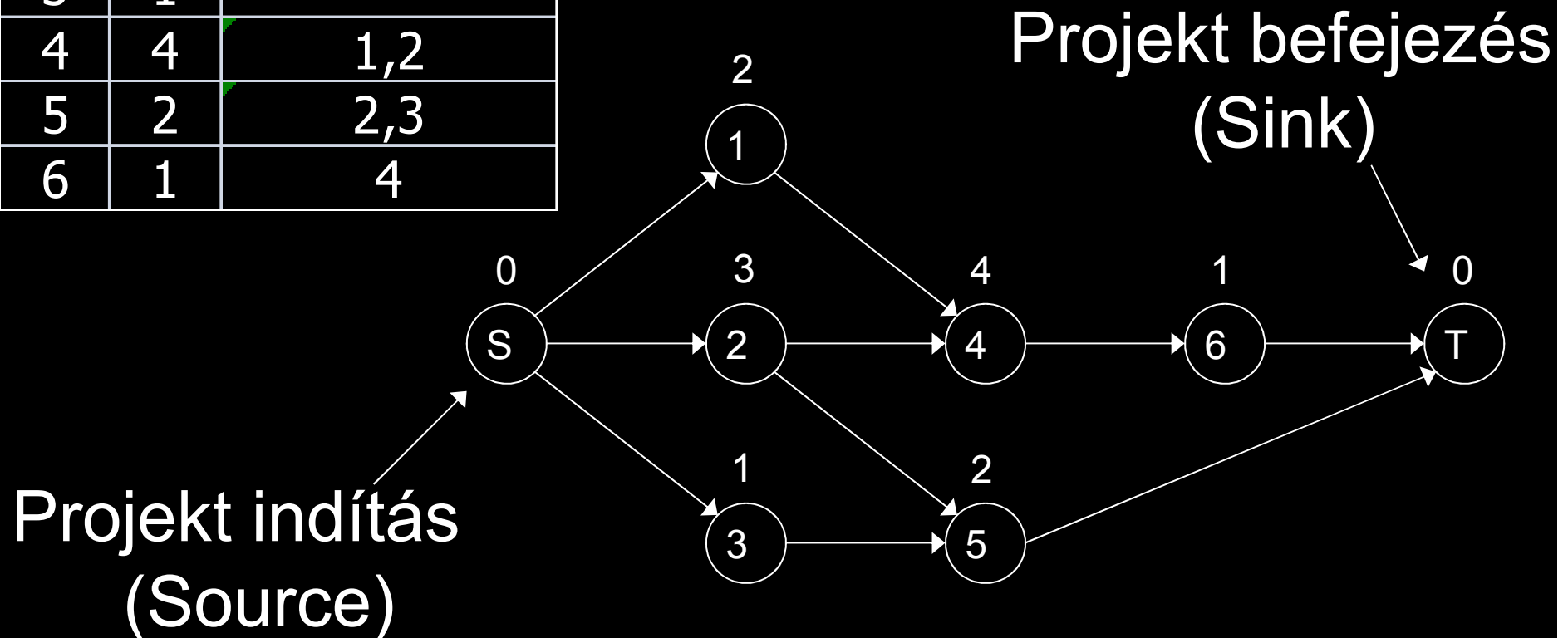
# Critical Path példa 1



# CPM példa 2

Feladat    Műveleti idő    Megelőző feladat(ok)

Job	p(j)	Predecessors
1	2	-
2	3	-
3	1	-
4	4	1,2
5	2	2,3
6	1	4

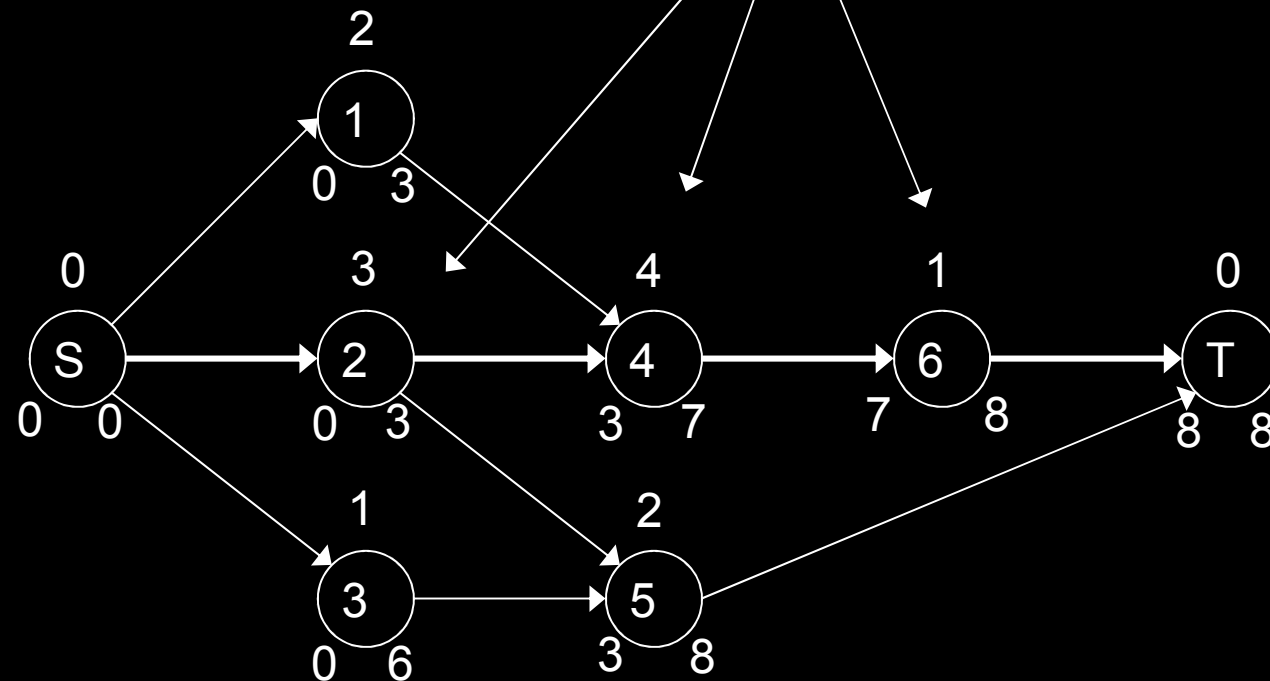
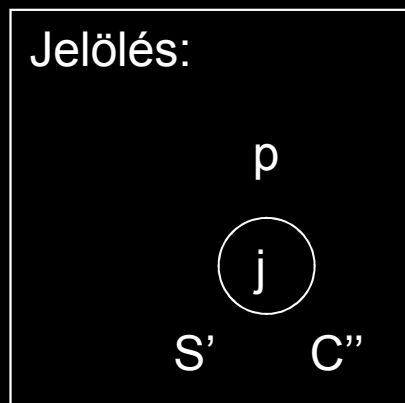


Job	p(j)	Predecessors	S'	C''
1	2	-	0	3
2	3	-	0	3
3	1	-	0	6
4	4	1,2	3	7
5	2	2,3	3	8
6	1	4	7	8

## CPM példa 2 (folyt.)

Kritikus feladat (Critical job):

$$S' + p = C' = C'' = S'' + p$$

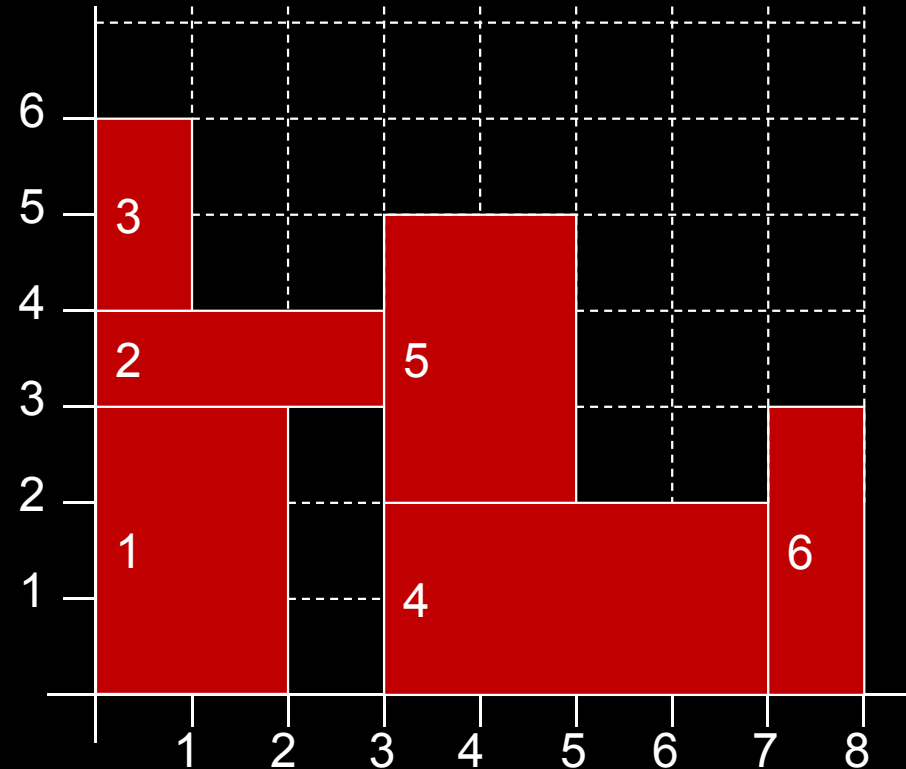


# Erőforrás korlátozott projektütemezés(RCPS)

- A munkák erőforrást igényelnek:

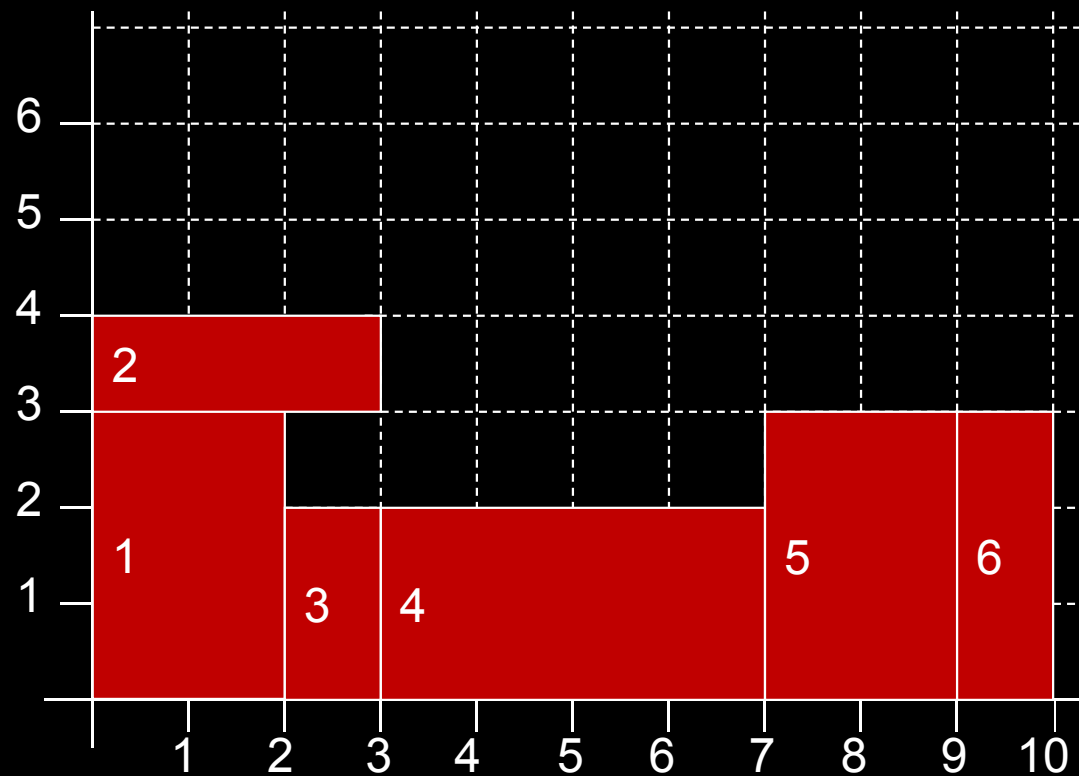
Job	p(j)	Predecessors	S'	C''	R(1,j)
1	2	-	0	3	3
2	3	-	0	3	1
3	1	-	0	6	2
4	4	1,2	3	7	2
5	2	2,3	3	8	3
6	1	4	7	8	3

↑  
Erőforrás-igény



# Erőforrás korlátozott projektütemezés (RCPS)

- Tételezzük fel, hogy  $R_I = 4$ , ekkor:



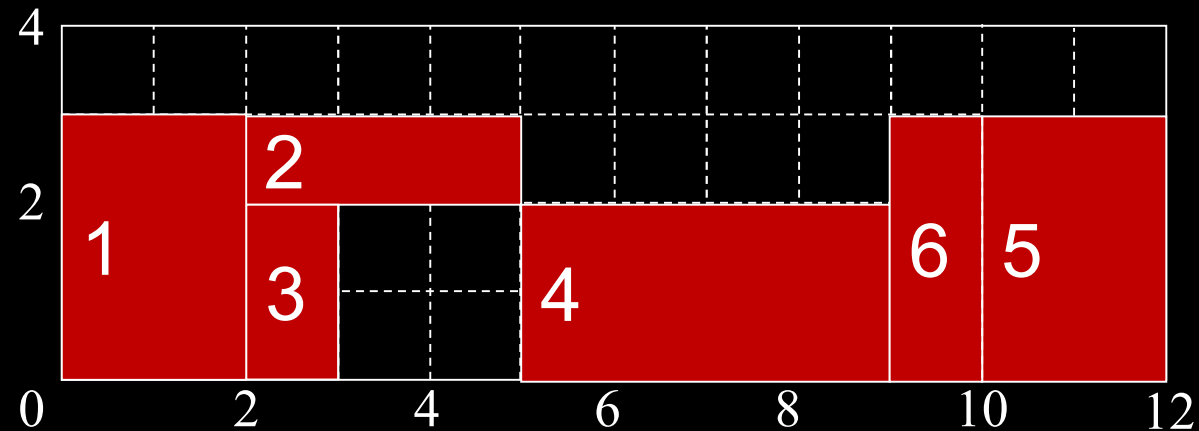
$C_{\max}$  nő 2 időegységgel!

# RCPSP

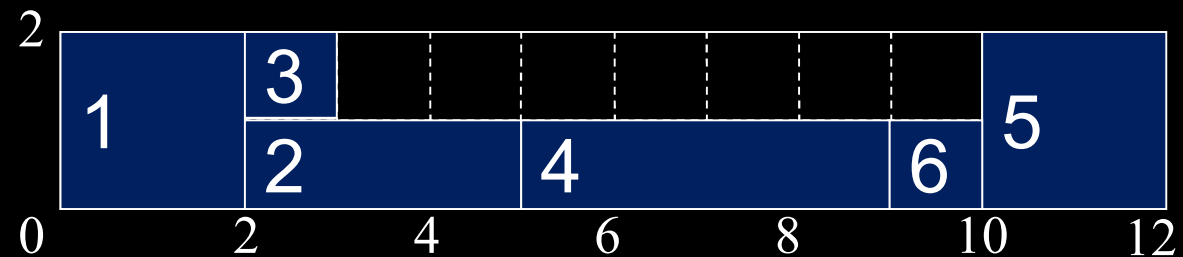
## Példa

Job	$p(j)$	$P(j)$	$S'$	$C''$	$R(1,j)$	$R(2,j)$
1	2	-	0	3	3	2
2	3	-	0	3	1	1
3	1	-	0	6	2	1
4	4	1,2	3	7	2	1
5	2	2,3	3	8	3	2
6	1	4	7	8	3	1

$$R_1 = 4$$



$$R_2 = 2$$





# Köszönöm a figyelmet!

Dr. Kulcsár Gyula  
Miskolci Egyetem Alkalmazott Informatikai Tanszék  
kulcsar@ait.iit.uni-miskolc.hu  
<http://ait.iit.uni-miskolc.hu/~kulcsar>

