Gráfalgoritmusok

Gaskó Noémi

2023. március 31.

Tartalomjegyzék

Egy feladat

Egy feladat

Hófehérkének egy szobát építenek a törpök.

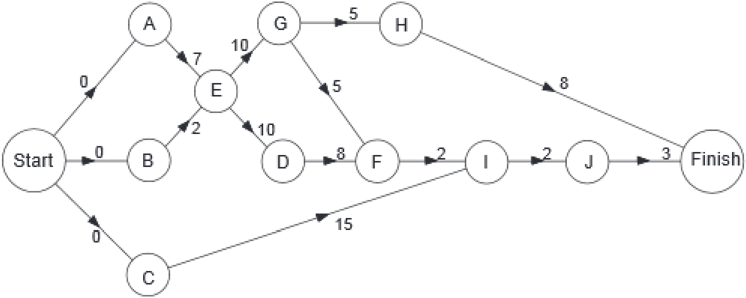
Milyen feladatok vannak és mennyi ideig tartanak?

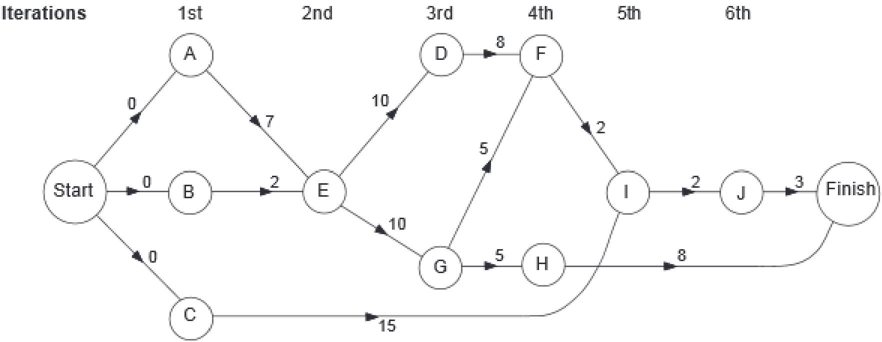
- alapot önteni - 7 nap - A
- ajtókeret elkészítése és elhelyezése - 2 nap - B
- kanalizálás - 15 nap - C
- szolgáltatások és kiegészítõk - 8 nap - D
- falak építése - 10 nap - E
- vakolás - 2 nap -F
- tetõ építése - 5 nap - G
- ablakok és ajtók - 8 nap - H
- csatornák - 2 nap - I
- kifesteni - 3 nap - J

Milyen sorrendben kell végrehajtani a feladatokat?

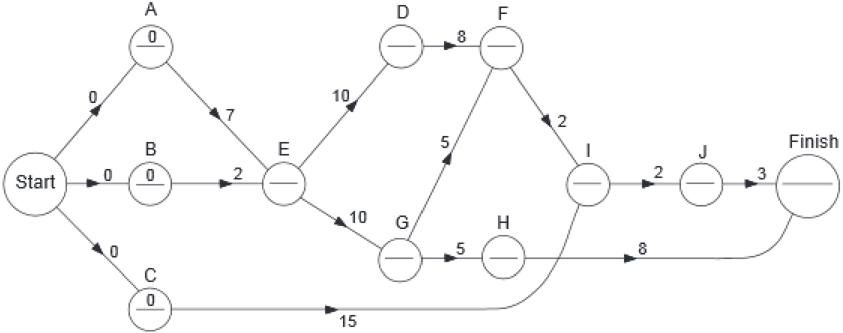
- D az E után kell következzen
- E az A és B után kell következzen
- F a D és G után kell következzen
- G az E után kell következzen
- H a G után kell következzen
- I a C és F után kell következzen
- J az I után kell következzen

A feladatunk:

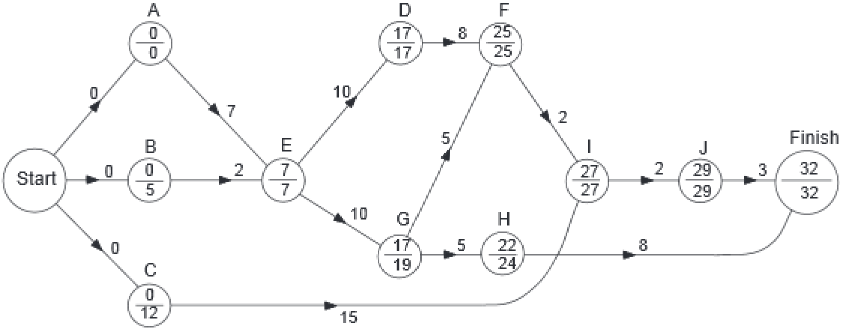



Iterations
1st
2nd
3rd
4th
5th
6th
A
D
8
F
0
7
10
2
5
Start
0
B
2
E
I
2
J
3
Finish
10
0
G
5
H
8
C
15

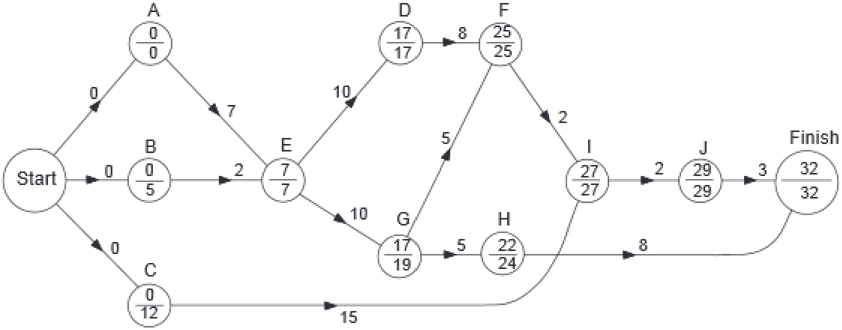
Mennyi ideig tart a munka?



Mennyi ideig tart a munka? (2)



Mennyi ideig tart a munka? (2)



Egy szimuláció: Egy példa

Hogy néz ki a feladat gráfmodellje?

- a tevékenységeket élekkel jelöljük, a csúcsok pedig az események

Hogy néz ki a feladat gráfmodellje?

- a tevékenységeket élekkel jelöljük, a csúcsok pedig az események
- a csúcsok felelnek meg a tevékenységeknek, az élek a tevékenységek közötti kapcsolatot jelölik

Kritikus utak - első modell

Tevékenységi gráf

A $G = (V, E, W)$ összefüggő, irányított, kört nem tartalmazó gráfot tevékenységi gráfnak nevezzük, ha:
az élek tevékenységeket jelölnek, az élekhez rendelt súlyok a tevékenységek végrehajtásához szükséges időt jelölik.

létezik egy kezdőcsúcs, amelybe egyetlen él sem fut
létezik egy végcsúcs, azaz egyetlen él sem fut ki belőle

Szintekre bontás

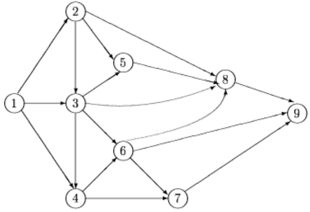
```
SzintekreBontás(G)
for i = 1 to n do
    l(i) := 1
for i = 1 to n do
    call Next(G, l, i)
return(l)
```

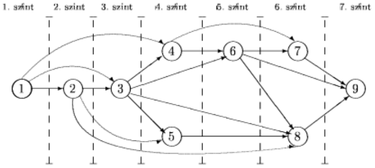
ahol

```
Next(G, l, i)
for j = 1 to n do

        if (v_i, v_j) ∈ E and (l(j) ≤ l(i)) then
            l(j) := l(i) + 1
            if j < i then call Next(G, l, j)
    return l
```

Szintekre bontás - példa





Az algoritmus

CPMél(G)
$t_1 := 0$
for $j = 2, 3, \ldots n$ **do**

$$t_j = max_{v_i \in N^{be}(v_j)}(t_i + d_{ij})$$

$t_n^* := t_n$
for $i = n-1, n-2 \ldots 1$ **do**

$$t_i^* = min_{v_j \in N^{ki}(v_i)}(t_j^* - d_{ij})$$

return t, t^*

t_i - a legkorábbi időpont,
t_i^* - a legkésőbbi időpont, amikor a tevékenység elkezdődhet
$d(i, j)$ - v_i és v_j közötti tevékenység időtartama

Időtartalékok

- teljes idõtartalék: $R_t(v_i, v_j) = t_j^* - t_i - d_{ij}$ - ennyi idõvel lehet késõbb kezdeni anélkül, hogy befolyásolja az egész feladat elvégzésének idõtartamát.

Időtartalékok

- teljes idõtartalék: $R_t(v_i, v_j) = t_j^* - t_i - d_{ij}$ - ennyi idõvel lehet késõbb kezdeni anélkül, hogy befolyásolja az egész feladat elvégzésének idõtartamát.
- szabad idõtartalék: $R_f(v_i, v_j) = t_j - t_i - d_{ij}$ - ennyi idõvel lehet késõbb kezdeni anélkül, hogy ez befolyásolja a t_j idõpontot.

Időtartalékok

- teljes idõtartalék: $R_t(v_i, v_j) = t_j^* - t_i - d_{ij}$ - ennyi idõvel lehet késõbb kezdeni anélkül, hogy befolyásolja az egész feladat elvégzésének idõtartamát.
- szabad idõtartalék: $R_f(v_i, v_j) = t_j - t_i - d_{ij}$ - ennyi idõvel lehet késõbb kezdeni anélkül, hogy ez befolyásolja a t_j idõpontot.
- biztos idõtartalék: $R_s(v_i, v_j) = max(t_j - t_i^* - d_{ij}, 0)$ - a (v_i, v_j) tevékenységet ennyi idõvel lehet késõbb befejezni anélkül, hogy ez befolyásolja az egész feladat elvégzésének idõtartamát.

Kritikus út

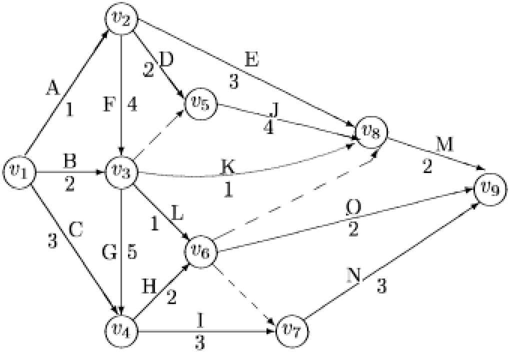
Kritikus út

Azok a tevékenységek, amelyek esetén mindhárom idõtartalék 0, a **kritikus úton** vannak, és **kritikus tevékenységnek** nevezzük õket.

Feladat

tevékenység	előző tevékenységek	időtartam
A	–	1
B	–	2
C	–	3
D	A	2
E	A	3
F	A	4
G	B, F	5
H	C, G	2
I	C, G	3
J	B, F, D	4
K	B, F	1
L	B, F	1
M	E, H, J, K, L	2
N	H, I, L	3
O	H, L	2

Feladat (2)



Feladat (3)

csúcs	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
t_i	0	1	5	10	5	12	13	12	16
t_i^*	0	1	5	10	10	13	13	14	16

Feladat (4)

tevékenység	időtartam	teljes idő-tartalék: R_t	szabad idő-tartalék: R_f	biztos idő-tartalék: R_s
A	1	0	0	0
B	2	3	3	3
C	3	7	7	7
D	2	7	2	2
E	3	10	8	8
F	4	0	0	0
G	5	0	0	0
H	2	1	0	0
I	3	0	0	0
J	4	5	3	0
K	1	8	6	6
L	1	7	6	6
M	2	2	2	0
N	3	0	0	0
O	2	2	2	1

Második modell

csúcsok: tevékenységek
élek: ezek egymásutánisága
kezdõ és végcsúcs: egy-egy fiktiv tevékenységnek felel meg.

$t_m(v_i)$	v_i	$t_m^*(v_i)$
$t_M(v_i)$	d_i	$t_M^*(v_i)$

- d_i a v_i tevékenység idõtartama
- $t_m(v_i)$ - legkorábbi idõpont, amikor a v_i tevékenység megkezdõdhet
- $t_m^*(v_i)$ - legkorábbi idõpont, amikor a v_i tevékenység befejezõdhet
- $t_M(v_i)$ - legkésõbbi idõpont, amikor a v_i tevékenység megkezdõdhet
- $t_M^*(v_i)$ - legkésõbbi idõpont, amikor a v_i tevékenység befejezõdhet

CPM él algoritmus

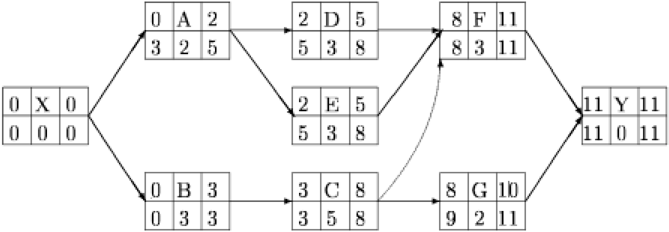
CPMcsúcs(D)
$t_m(v_1) = 0$
$t_m^*(v_1) = d_1$
for $j = 2, 3 \ldots, n$ **do**
    $t_m(v_j) = max_{v_i \in N^{be}(v_j)} t_m^*(v_i)$
    $t_m^*(v_j) = t_m(v_j) + d_j$
$t_M^*(v_n) = t_m^*(v_n)$
$t_M(v_n) = t_M^*(v_n) - d_n$
for $i = n-1, n-2, \ldots, 1$ **do**
    $t_M^*(v_i) = min_{v_j \in N^{ki}(v_i)} t_M(v_j)$
    $t_M(v_i) = t_M^*(v_i) - d_i$
return $t_m, t_m^*, t_M, t_M^*,$

Egy tevékenység kritikus, ha $t_m(v) = t_M(v)$ (és $t_m^*(v) = t_M^*(v)$).

Feladat

tevékenység	előző teveékenységek	időtartam
A	–	2
B	–	3
C	B	5
D	A	3
E	A	3
F	C, D, E	3
G	C	2

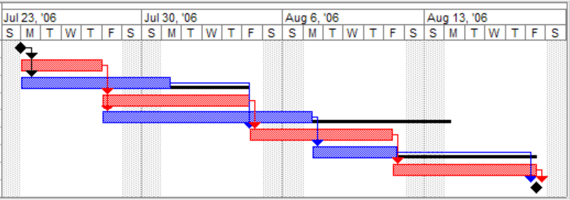
Feladat (2)


0
X
0
0
0
0
0
A
2
3
2
5
0
B
3
0
3
3
2
D
5
5
3
8
2
E
5
5
3
8
3
C
8
3
5
8
8
F
11
8
3
11
8
G
10
9
2
11
11
Y
11
11
0
11

Alkalmazások

Projekt management:

ID	Task Name	Predecessors	Duration
1	Start		0 days
2	a	1	4 days
3	b	1	5.33 days
4	c	2	5.17 days
5	d	2	6.33 days
6	e	3,4	5.17 days
7	f	5	4.5 days
8	g	6	5.17 days
9	Finish	7,8	0 days



PERT módszer

- Program Evaluation and Review Technique

PERT módszer

- Program Evaluation and Review Technique
- elõször: az Egyesült Államok haditengerészeténél 1957-ben rakétaprogram irányítására alkalmazták, segítségével 5 év helyett 3,5 év alatt hajtották végre a programot

PERT módszer

- Program Evaluation and Review Technique
- elõször: az Egyesült Államok haditengerészeténél 1957-ben rakétaprogram irányítására alkalmazták, segítségével 5 év helyett 3,5 év alatt hajtották végre a programot
- eseménybeállítottságú eljárás

PERT módszer

- Program Evaluation and Review Technique
- elõször: az Egyesült Államok haditengerészeténél 1957-ben rakétaprogram irányítására alkalmazták, segítségével 5 év helyett 3,5 év alatt hajtották végre a programot
- eseménybeállítottságú eljárás
- háromféle idõt használ: optimista, pesszimista, legvalószínûbb