

Mrežno planiranje

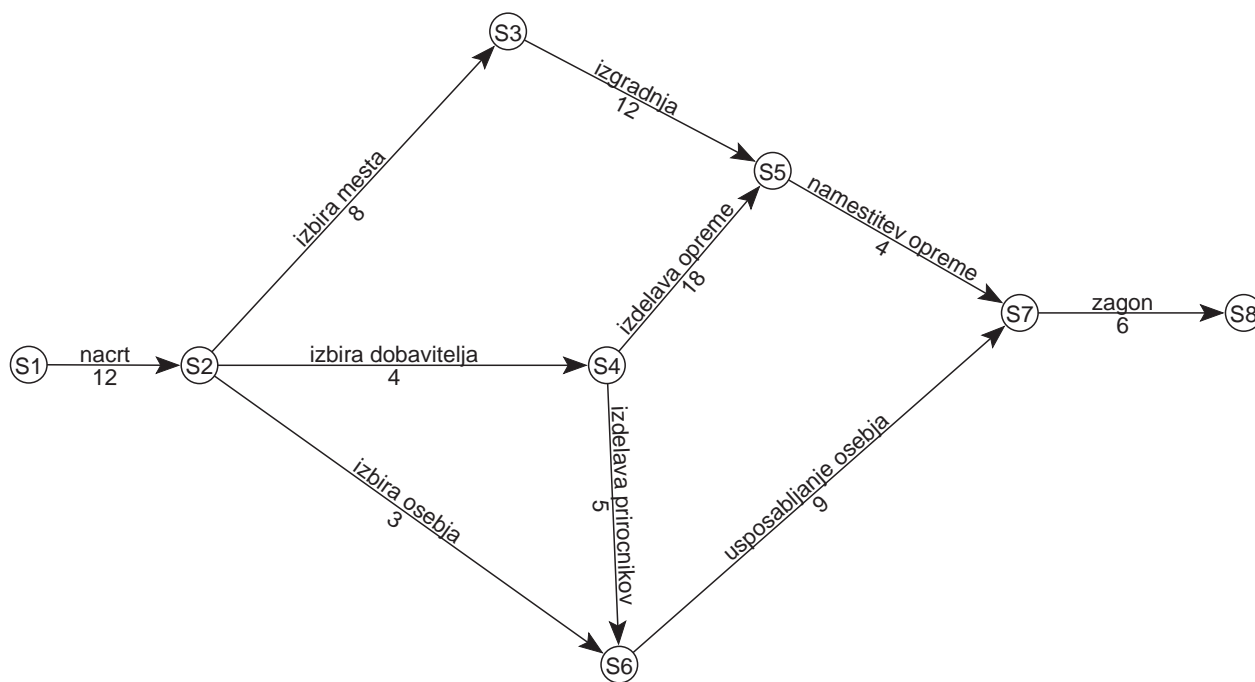
Predpostavimo, da je neko delo (projekt) sestavljeno iz večih opravil, nekatera od teh lahko opravljamo vzporedno, nekatera pa lahko začnemo šele, ko so končana nekatera druga opravila.

Npr. pri gradnji hiše lahko istočasno napeljujemo vodovodno in električne instalacije, ne moremo pa graditi strehe, dokler ni hiša sezidana.

Vsako opravilo zahteva določen čas.

Naloga: Koliko časa potrebujemo da opravimo celotno delo?

Predstavitev z **grafom** – primer izgradnja toplarne:



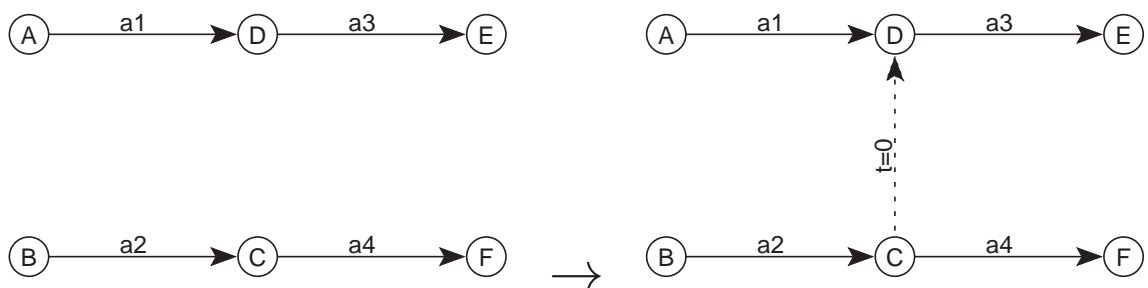
Zaporedna in vzporedna opravila

Zaporedni opravila sta npr: *izbira mesta* - *izgradnja* (izgradnja se ne more začeti dokler mesto ni izbrano).

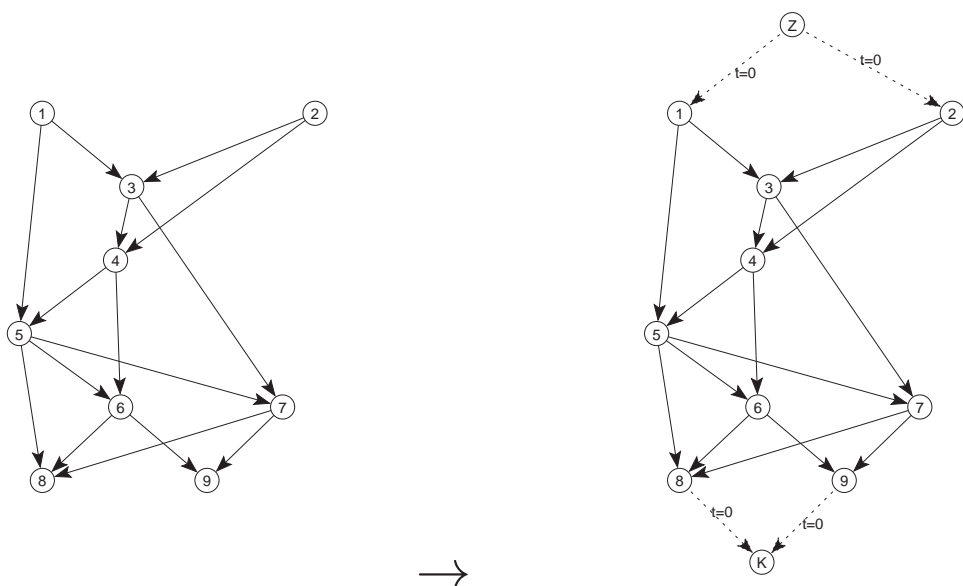
Vzporedni opravila sta npr: *izbira dobavitelja* in *izbira osebja* (dobavitelje in osebje lahko neodvisno izbiramo istočasno).

Predpostavke:

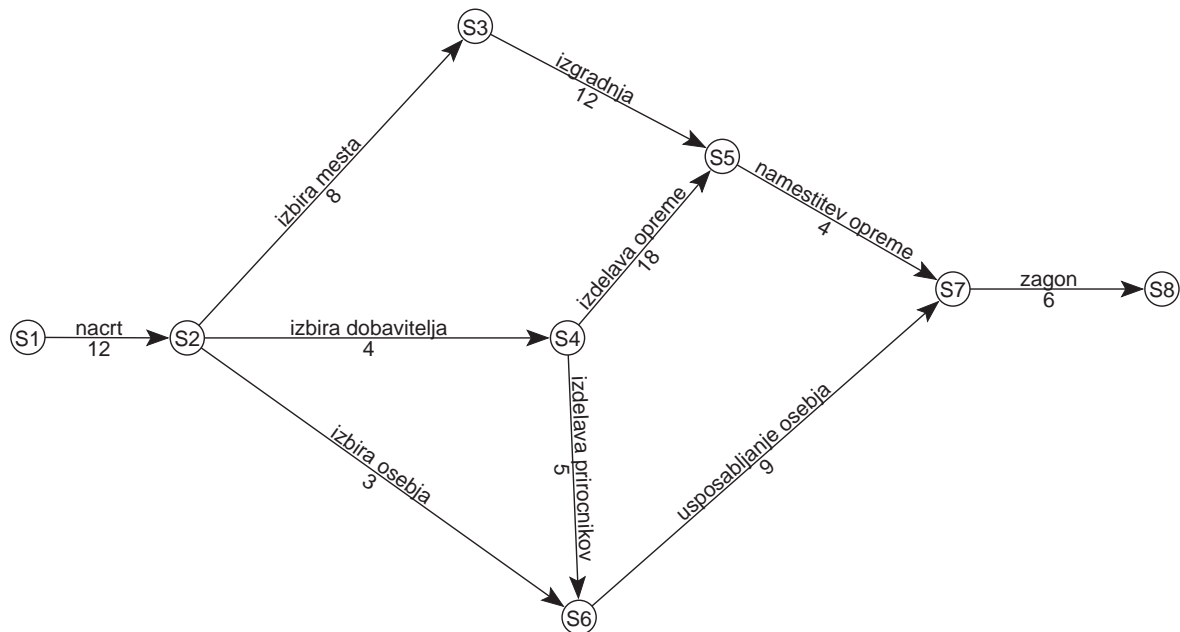
1. Če se opravilo a_3 ne more začeti dokler ni končano opravilo a_2 , čeprav med stanjema C in D ni nobene povezave, dodamo **navidezno opravilo** s časom izvajanja $t = 0$.



2. Predpostavimo, da imamo samo eno **začetno stanje** (stanje v katero ne vodi nobena povezava) in samo eno **končno stanje** (stanje iz katerega ne vodi nobena povezava). Če imamo več začetnih in končnih stanj, dodamo **enolični začetek** oz **enolični konec**:



Primer: izgradnja toplarne:



Naloga: Koliko časa rabimo, da opravimo vsa opravila?

1. Graf je acikličen. Za aciklične grafe velja, da če začnemo pot v katerikoli točki in sledimo smerem povezav, se na noben način ne moremo vrniti nazaj v točko, ampak vedno pridemo v eno od končnih točk. Zato lahko za manjše grafe pregledamo vse poti od začetne do končne točke. V našem primeru:

$$S1-S2-S3-S5-S7-S8: 12+8+12+4+6 = 42$$

$$\boxed{S1-S2-S4-S5-S7-S8}: 12+4+18+4+6 = \boxed{44}$$

$$S1-S2-S4-S6-S7-S8: 12+4+5+9+6 = 36$$

$$S1-S2-S6-S7-S8: 12+3+9+6 = 30$$

2. V primeru večjega števila opravil pregled vseh možnih poti seveda ne pride v poštev. Ena od metod, ki omogoča, da hitro pridemo do rešitve, je CPM.

Metoda kritične poti

CPM – Critical Path Method

Za vsako opravilo (povezavo) želimo določiti:

- Kdaj najprej se lahko začne (*začetek.min*)?
- Kdaj najkasneje se mora začeti (*začetek.max*)?
- Kdaj najprej se lahko konča (*konec.min*)?
- Kdaj najkasneje se mora končati (*konec.max*)?

Če hočemo izračunati omenjene čase, moramo prej za vsako stanje (točke v grafu) določiti:

1. **ZAROK:** Kdaj so vsa opravila s koncem v danem stanju končana? (prej se opravila z začetkom v tem stanju ne morejo začeti)
2. **KOROK:** Kdaj najkasneje moramo priti v dano stanje, da ne zakasnimo celotnega projekta?

Naredimo tabelo vseh stanj: (Excel)

Algoritem za izračun zaroka (gremo od začetnega stanja v smeri puščic):

Postavi začetno točko v seznam in ji priredi zarok:=0

Dokler obstaja vsaj ena točka v seznamu **ponavljaj**

Za vsa opravila o z začetkom v tej točki **ponovi**

$\text{max} := o(\text{začetek}).\text{zarok} + o(\text{trajanje})$

Če je $o(\text{konec}).\text{zarok} < \text{max}$ **potem**

$o(\text{konec}).\text{zarok} := \text{max}$

$o(\text{konec}).\text{prednik} := \text{začetek}$

Če so bila vsa opravila, ki vodijo v konec obdelana, **potem**

Postavi konec na konec seznama

Algoritem za izračun koroka (gremo od končnega stanja v nasprotni smeri kot kažejo puščice):

Postavi konec v seznam in mu priredi korok:=zarok

Dokler obstaja vsaj ena točka v seznamu **ponavljaj**

Za vsa opravila o s koncem v tej točki **ponovi**

$\text{min} := o(\text{konec}).\text{korok} - o(\text{trajanje})$

Če je $o(\text{začetek}).\text{korok} > \text{min}$ **potem**

$o(\text{začetek}).\text{korok} := \text{min}$

Če so bila vsa opravila, ki imajo ta začetek obdelana, **potem**

Postavi začetek na konec seznama

Izračunamo tabelo vseh opravil: (Excel)

Celotno dovoljeno zamujanje: Koliko lahko največ zamudimo pri izbranem opravilu, da ne zakasnimo celotnega projekta – celoten projekt lahko še vedno končamo v enakem času.

Celotno dovoljeno zamujanje:=

konec.korok - začetek.zarok - trajanje

Prosto dovoljeno zamujanje: Koliko lahko največ zamudimo pri izbranem opravilu, da se vsa opravila, ki sledijo danemu opravilu še vedno lahko začnejo po izračunanem času začetka.

Prosto dovoljeno zamujanje:=

konec.zarok - začetek.zarok - trajanje

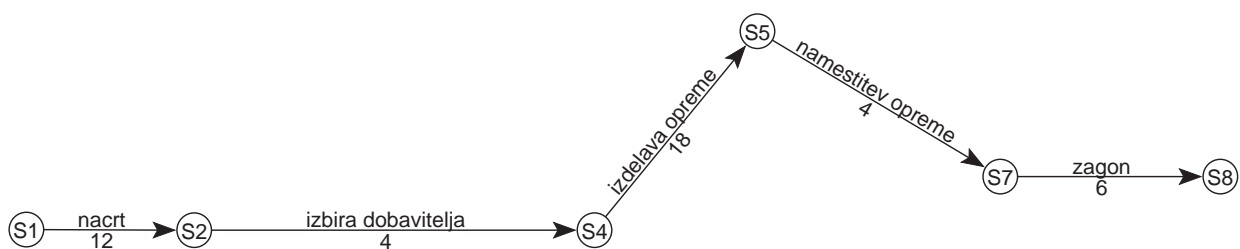
Najpomembnejši rezultat metode CPM pa je **kritična pot**, to je zaporedje opravil, ki določa trajanje celotnega projekta. Na tej poti niso dovoljena nikakršna zamujanja, ker bi le ta povzročila tudi zakasnitev celotnega projekta. Za vsako opravilo o na kritični poti velja:

$$o(\text{začetek}).\text{min} = o(\text{začetek}).\text{max}$$

$$o(\text{konec}).\text{min} = o(\text{konec}).\text{max}$$

$$o(\text{konec}).\text{min} = o(\text{začetek}).\text{min} + o(\text{trajanje})$$

Kritična pot za primer toplarne:

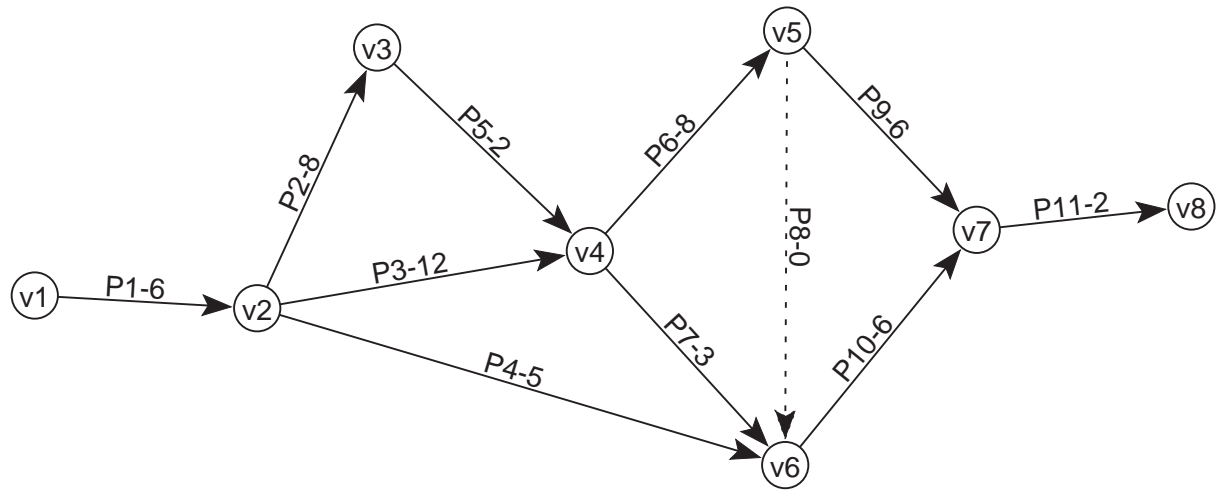


Naloga1

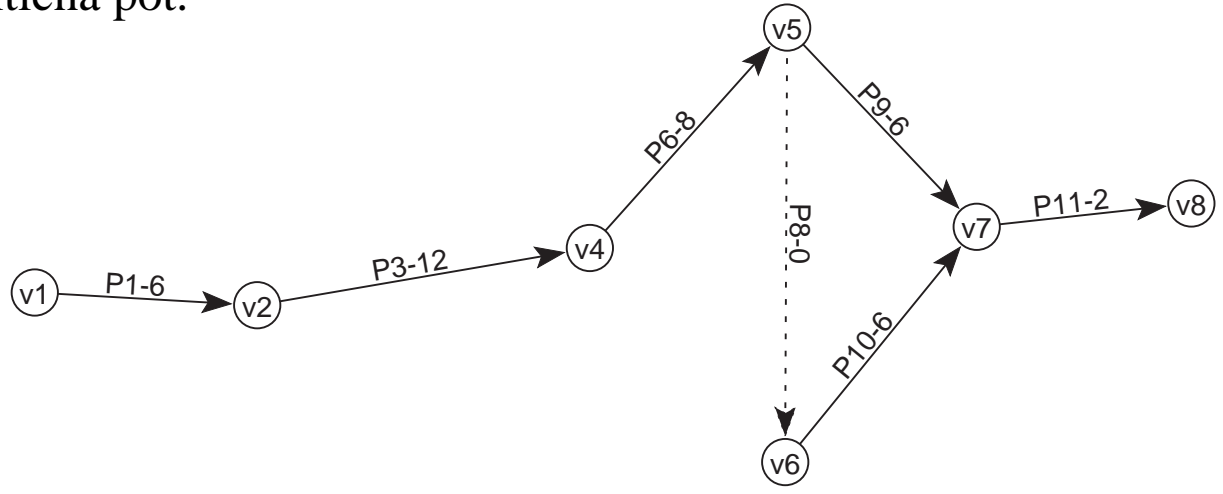
Za podano omrežje narišite sliko, poiščite kritično pot ter za vsako opravilo izračunajte celotno in prosto dovoljeno zamujanje.

Opravilo	Predhodno morajo biti opravljena	Trajanje
P1	∅	6
P2	P1	8
P3	P1	12
P4	P1	5
P5	P2	2
P6	P3, P5	8
P7	P3, P5	3
P8	P6	0
P9	P6	6
P10	P4, P7, P8	6
P11	P9, P10	2

Rešitev:
Slika omrežja.

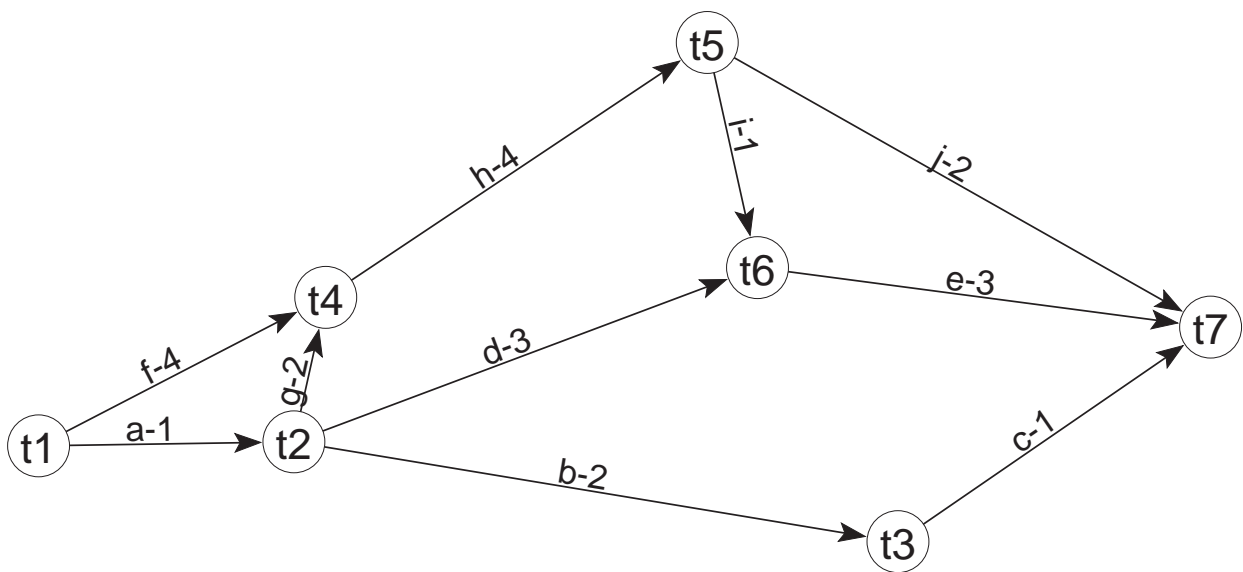


Kritična pot.



Naloga2

Za podano omrežje poiščite kritično pot ter za vsako opravilo izračunajte celotno in prosto dovoljeno zamujanje.



Rešitev: kritična pot.

