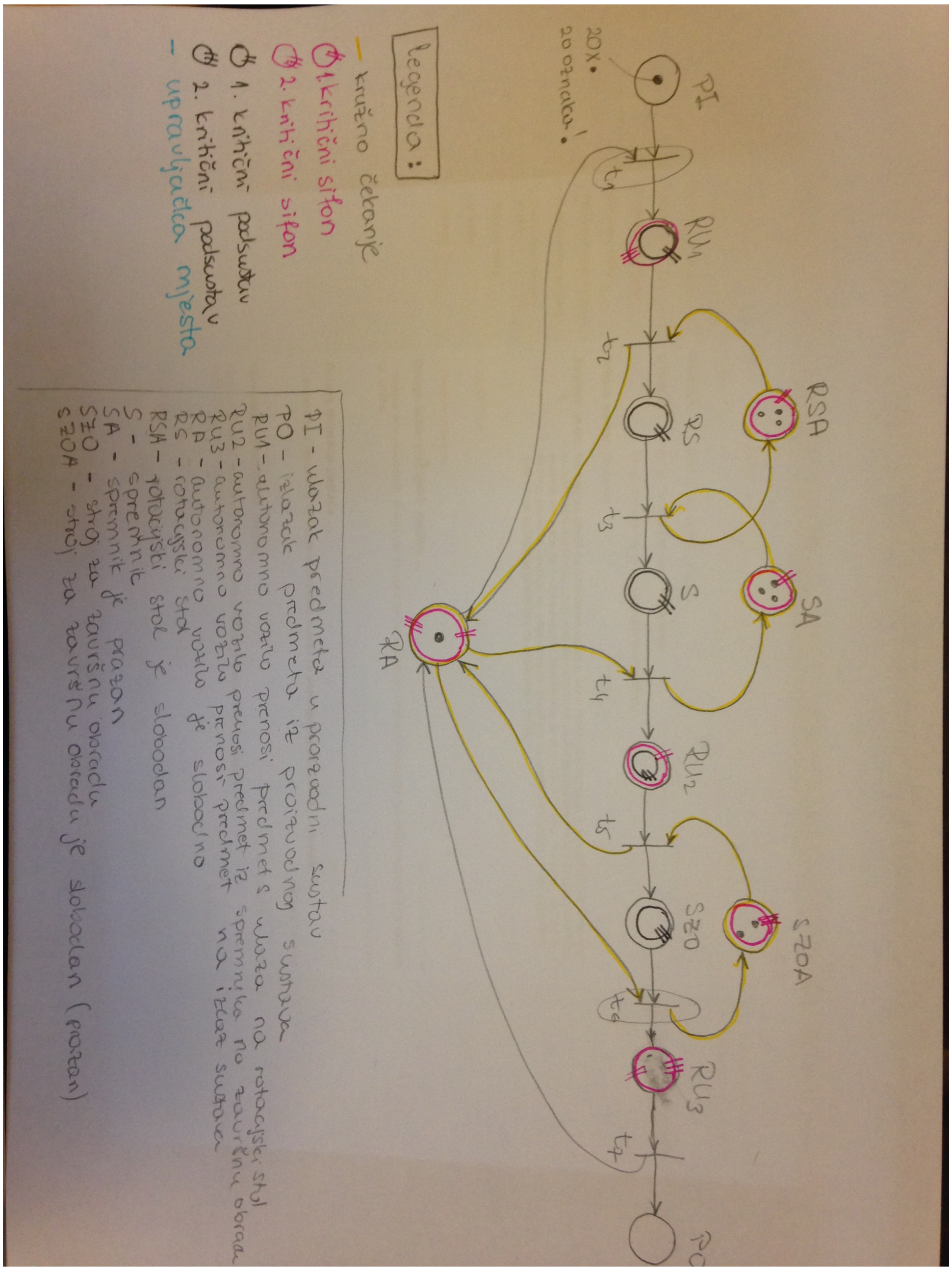
2. Domaća zadaća

Petrijeva mreža i Bankarev algoritam



2. Domaća zadaća

1. PROIZVODNA LINIJA:

- ULAZNA STANICA (UL) predmeta
- IZLAZNA STANICA (IZ) predmeta
- AUTONOMNOG VOZILA (AGV)
- ROTACIJSKOG STOLA → ZA BUŠENJE RUPA
  ↳ KAPACITETA 3 predmeta
- SPREMNIK ZA PROBUŠENE PREDMETE
  ↳ kapaciteta 3S
- STROJ ZA ZAVRŠNU OBRADU
  ↳ kapaciteta 2

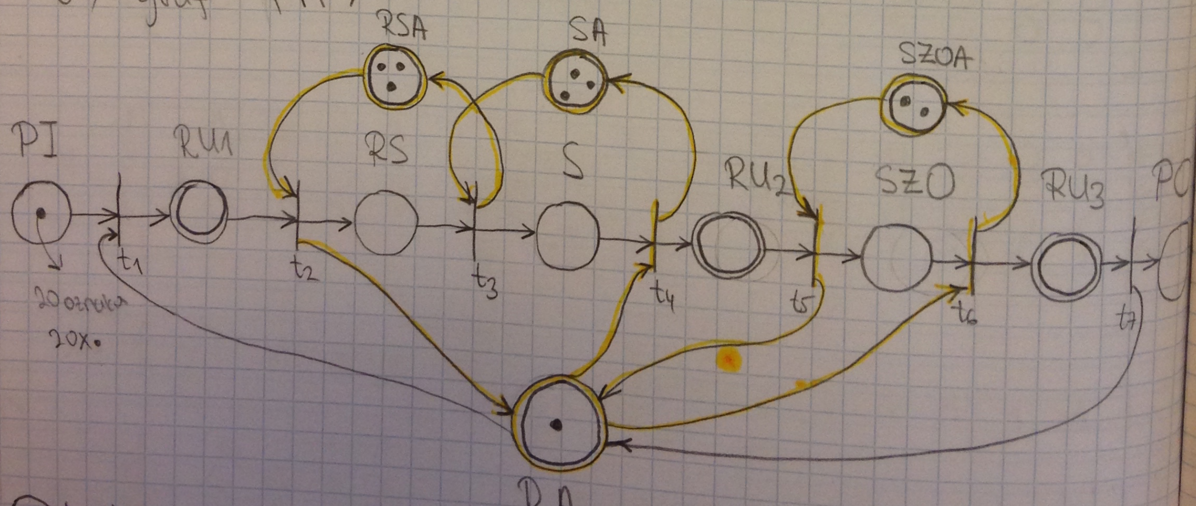
PROLAZ:

UL → AGV → RS → S → AGV → SZO →
→ AGV → IZ

↳ između resursa → ne postoje dodatni spremnici za smještaj

ulaz: 20 predmeta

a) graf PM:



kritični sifon
– Postoji složeno kružno čekanje

b) kružna čekanja i kritični podsustav

$C_1 = \{ RA, RSA, SA \}$

$C_2 = \{ RA, SZOA \}$

ulazni i izlazni prijelazi u C_1:

1. $\bullet C_1 = \{ t_2, t_3, t_4, t_5, t_7 \} = T_{C1}^i$

$C_1 \bullet = \{ t_1, t_2, t_3, t_4, t_6 \}$

$T_{C1} = \bullet C_1 \cap C_1 \bullet = \{ t_2, t_3, t_4 \}$

prijelazi koji nisu blokirani resursima u C_1:

$$T_{S1} = T_{C1}^i \setminus T_{C1} = \{ t_2, t_3, t_4, t_5, t_7 \} \setminus \{ t_2, t_3, t_4 \} =$$
$$= \{ t_5, t_7 \}$$

– skup poslova sifona $J_S(C_1)$ preko poslova kružnog čekanja $J(C_1)$:

$\bullet T_{S1} = \{ RU_2, RU_3 \}$

$J(C_1) = \{ RU_1, RS, S, RU_2, RU_3 \}$ poslovi kružnog čekanja

$$J_S(C_1) = \bullet T_{S1} \cap J(C_1) =$$
$$= \{ RU_2, RU_3 \}$$

1. KRITIČNI SIFON ⇒

$$S_{C1} = C_1 \cup J_S(C_1) =$$
$$= \{ RA, RSA, SA \} \cup \{ RU_2, RU_3 \} = \{ RA, RSA, SA, RU_2, RU_3 \}$$

② $C_2 = \{RA, SZOA\}$

$$\cdot C_2 = \{t_2, t_5, t_6, t_7\} = T_{c_2}^i$$

$$C_2 \cdot = \{t_1, t_4, t_5, t_6\}$$

$$T_{c_2} = \cdot C_2 \cap C_2 \cdot = \{t_5, t_6\}$$

ulazni prijelazi koji nisu blokirani resursima C_2

$$T_{s_2} = T_{c_2}^i \setminus T_{c_2} = \{t_2, t_5, t_6, t_7\} \setminus \{t_5, t_6\} = \{t_2, t_7\}$$

skup poslova sifona $J_S(C_2)$ preko poslova kružnog čekanja:

$$\cdot T_{s_2} = \{RU_1, RU_3\}$$

$$J(C_2) = \{RU_1, RU_2, SZO, RU_3\}$$ poslovi kružnog čekanja

$$J_S(C_2) = \cdot T_{s_2} \cap J(C_2) = \{RU_1, RU_3\} \cap \{RU_2, SZO, RU_3, RU_1\} = \{RU_1, RU_3\}$$

2. KRITIČNI SIFON:

$$S_{c_2} = C_2 \cup J_S(C_2) = \{RA, SZOA\} \cup \{RU_1, RU_3\} = \{RA, SZOA, RU_1, RU_3\}$$

1. kritični podsustav:

$C_1^\bullet = T_{C_1}^\circ$

$T_{Q_1} = T_{C_1}^\circ \setminus T_{C_1} = \{t_1, t_2, t_3, t_4, t_6\} \setminus \{t_2, t_3, t_4\} =$

$= \{t_1, t_6\}$

$T_{Q_1}^\bullet \cap J(C_1) = J_Q(C_1) = \{RU_1, RU_3\} \cap \{RU_1, RS, S, RU_2, RU_3\}$

$= \{RU_1, RU_3\}$ skup poslova zamke

$Q_{C_1} = C_1 \cup J_{Q_1}(C_1) = \{RA, RSA, SA\} \cup \{RU_1, RU_3\} =$

$= \{RA, RSA, SA, RU_1, RU_3\}$ kritična zamka

$J_N(C_1) = J(C_1) \setminus \{J_Q(C_1) \cup J_S(C_1)\} =$ skup neutralnih poslova

$= \{RU_1, RS, S, RU_2, RU_3\} \setminus \{RU_1, RU_2, RU_3\} =$

$= \{RS, S\}$

$J_{SQ}(C_1) = J_Q(C_1) \cap J_S(C_1) = \{RU_1, RU_3\} \cap \{RU_2, RU_3\} = \{RU_3\}$

$J_O(C_1) = \{J_Q(C_1) \setminus J_{SQ}(C_1)\} \cup J_N(C)$ kritični podsustav

$J_O(C_1) = \{\{RU_1, RU_3\} \setminus \{RU_3\}\} \cup \{RS, S\} =$

$= \{RU_1, RS, S\}$ 1. kritični podsustav!

VRIJEDI:

$J(C_1) = J_S(C_1) \cup J_O(C_1) = \{RU_1, RS, S, RU_2\}$

$m\{J(C_1)\} = m\{J_S(C_1)\} + m\{J_O(C_1)\}$ broj oznaka konstantan

2 kritični podsustav

$C_2^\circ = T_{C_2}^\circ = \{t_1, t_4, t_5, t_6\}$

$T_{Q_2} = T_{C_2}^\circ \setminus T_{C_2} = \{t_1, t_4, t_5, t_6\} \setminus \{t_5, t_6\} = \{t_1, t_4\}$

$T_{Q_2}^\bullet \cap J(C_2) = \{RU_1, RU_2\} \cap \{RU_1, RU_2, SZO, RU_3\} =$

$J_{Q_2}(C_2) = \{RU_1, RU_2\}$ skup poslova zamke

$Q_{C_2} = C_2 \cup J_{Q_2}(C_2) = \{RA, SZOA\} \cup \{RU_1, RU_2\}$

$= \{RA, SZOA, RU_1, RU_2\}$ kritična zamka

$J_N(C_2) = J(C_2) \setminus \{J_Q(C_2) \cup J_S(C_2)\} =$

$= \{RU_1, RU_2, SZO, RU_3\} \setminus \{RU_1, RU_2, RU_3\}$ skup neutralnih poslova

$= \{SZO\}$

$J_{SQ}(C_2) = J_Q(C_2) \cap J_S(C_2) = \{RU_1, RU_2\} \cap \{RU_1, RU_3\} = \{RU$

$J_O(C_2) = \{J_Q(C_2) \setminus J_{SQ}(C_2)\} \cup J_N(C_2) =$

$= \{\{RU_1, RU_2\} \setminus \{RU_1\}\} \cup \{SZO\} =$

$= \{RU_2, SZO\}$ 2. kritični podsustav

VRIJEDI:

$J(C_2) = J_S(C_2) \cup J_O(C_2) = \{RU_1, RU_3\} \cup \{RU_2, SZO\}$

$= \{RU_1, RU_2, RU_3, SZO\}$ //

$m\{J(C_2)\} = m\{J_S(C_2)\} + m\{J_O(C_2)\}$ br oznaka konstantan

c) odrediti $\boxed{\text{ako-onda}}$ pravila → sprječavanje zaglavljenj

– konflikte riješiti prema FBFS načelu

→ ako sifoni $S_{c1} = \{RA, RSA, SA, RU_2, RU_3\}$ i $S_{c2} = \{RA, SZOA, RU_1, R$ izgube sve oznake koje posjeduju inicijalno tada je sustav zaglavljen.

→ upravljačkim algoritmom možemo izbjeći zaglavljenja spriječiti da nastane konflikt

→ upravljačkim algoritmom potrebno je osigurati da u kritičnim sifonima ima uvijek barem jedna oznaka.

$$m_0(C_1) = m_0(S_{c1}) + m(J_0(C_1))$$

$$m_0(C_2) = m_0(S_{c2}) + m(J_0(C_2))$$

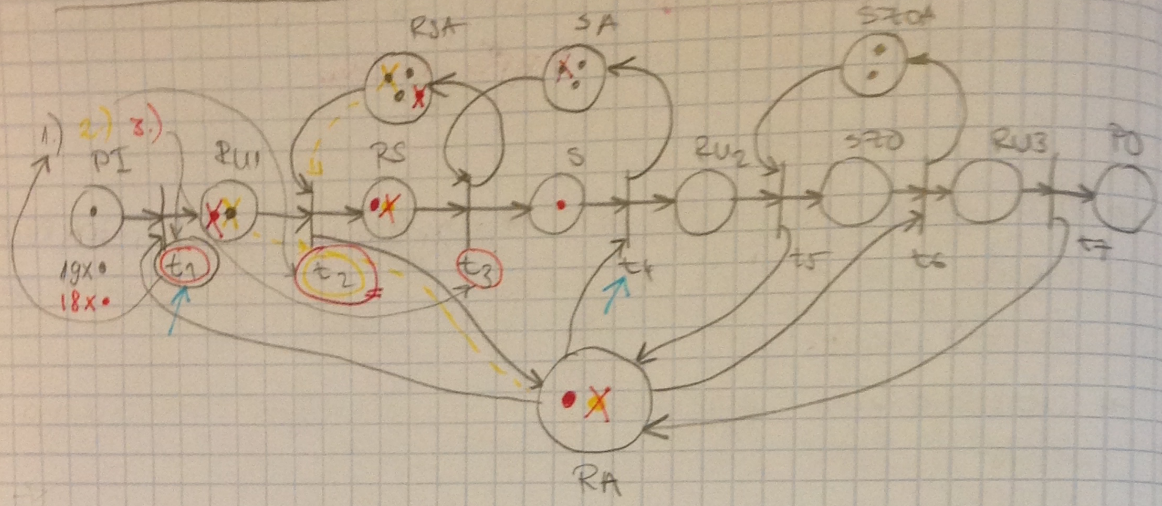
$$\underbrace{m_0(C_1) = 7} \qquad \underbrace{m_0(C_2) = 3}$$

Iz ovog slijedi da $m(J_0(C_1))$, odnosno $m(J_0(C_2))$ smije najviše imati 6, odnosno 2 oznake jer barem po jedna oznaka mora biti i u sifonima.

Obzirom da je $RU_2 \in S_{c1}$, ali i $RU_2 \in J_0(C_2)$ broj oznaka kritičnog podsustava $J_0(C_2)$ može biti i veći od broja 2, ako su te oznake pristigle iz sifona S_{c1}.

→ pražnjenjem sifona $\boxed{m_0(S_{c1}) = 0}$ i $\boxed{m_0(S_{c2})} = 0$ → sve oznake koje su u početnom stanju bile u kritičnom sifonu dospjele su u kritični podsustav što uzrokuje $\boxed{\text{ZAGLAVLJENJE}}$!!!

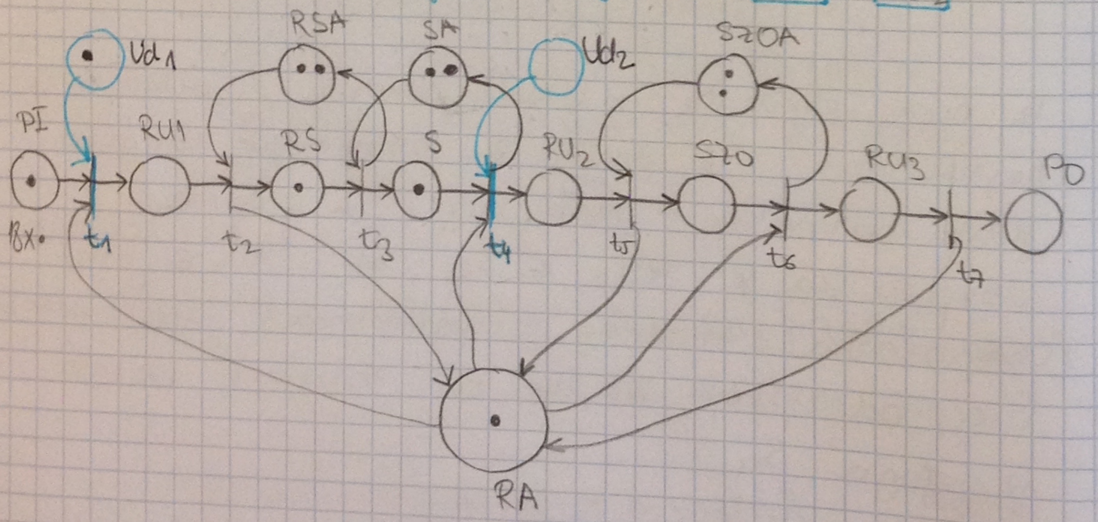
SPRJEČAVANJE ZAGLAVLJENJA → AKO-ONDA PRAVILA



→ u konfliktu su se našli prijelaz $\boxed{(t_1, t_4)}$ već nakon 4 okidanja prijelaza, sada je autonomni resurs slobodan a prijelazi t_1 i t_4 su omogućeni → upravljačkim algoritmom potrebno je odrediti što će prvo resurs poslužiti

→ prema načelu FBFS novija oznaka u sustavu ima prednost

↳ dodana su upravljačka mjesta! $\boxed{Ud_1}$ i $\boxed{Ud_2}$



Upravljački algoritam:

$\boxed{\text{AKO } (RA = 1 \wedge S \,!= 3 \wedge PI \neq 0) \text{ ONDA } (Ud_2 = 0,\ Ud_1 = 1)}$

⇒ postavljanjem oznake u Ud_1 znamo da će prvo propaliti prijelaz t_1, a obzirom da u Ud_2 nema oznake prijelaz t_4 ne može propaliti i time smo osigurali rješavanje konflikta.//

⇒ ali daljnjim praćenjem prijelaza došlo je do konflikta i između:

$\{t_1, t_6\}$

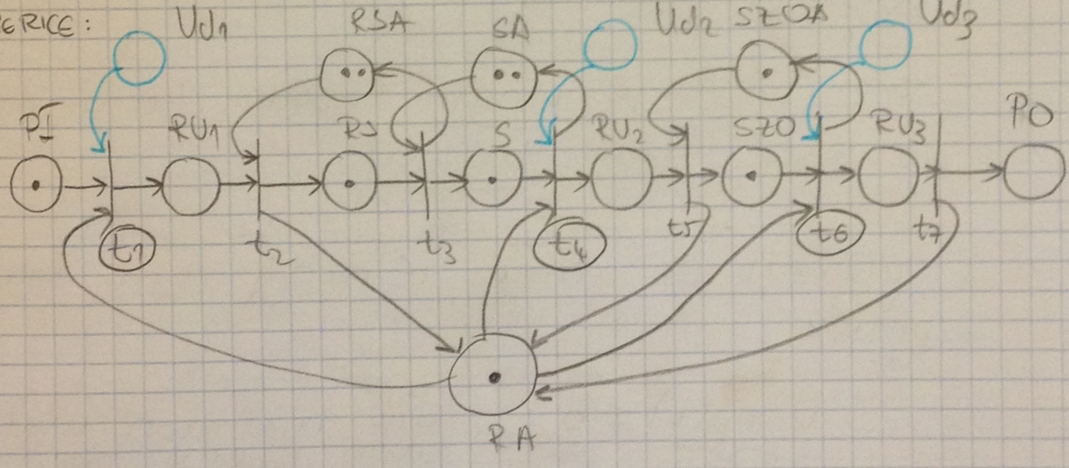
$\{t_4, t_6\}$

$\{t_1, t_4, t_6\}$

$\{t_1, t_4\}$

⇒ prema tome uvodi se još jedno upravljačko mjesto:

PRIMJERICE:



NOVA PRAVILA ZA RJ KONFLIKATA I ZAGLAVLJENJA!

AKO JE $(RA == 1 \ \&\& \ S != 3 \ \&\& \ PI != 0)$ ONDA
$(Ud_1 = 1 \ \&\& \ Ud_2 = 0 \ \&\& \ Ud_3 = 0)$

AKO JE $(RA == 1 \ \&\& \ S == 3 \ \&\& \ SZO < 2)$ ONDA
$(Ud_1 = 0 \ \&\& \ Ud_2 = 1 \ \&\& \ Ud_3 = 0)$

AKO JE (RA == 1 && SZO == 2) ONDA JE
($Ud_3 = 1$ && $Ud_2 = 0$ && $Ud_1 = 0$)

AKO JE (PI == 0 && RA == 1 && S > 0 && SZO < 2)
ONDA JE ($Ud_1 = 0$ && $Ud_2 = 1$ && $Ud_3 = 0$)

AKO JE (PI == 0 && RA == 1 && S == 0 && SZO ...
ONDA JE ($Ud_3 = 1$ && $Ud_2 = 0$ && $Ud_1 = 0$)

d) P-invarijantno upravljanje kojim se ukupan br predmeta u resursima RS, S, SZO ograničava na 5.
POSTUPAK SINTEZE. (KONFLIKT ⇒ LBFS NAČELO)

$$Wp = 0$$

P-invarijanta je binarni P-vektor (vektor mjesta) za koje vrijedi $\boxed{Wp = 0}$

Ograničiti br. oznaka u nekom dijelu Petrijeve mreže:

$\underbrace{L \cdot m_p(k)}_{\text{skupina mjesta}} \le \underbrace{b}_{\text{br oznaka}}$ Skup nejednakosti / uvjeta koji govori koliki je dopušteni br maksimalno oznaka u skupinama mjesta

⇒ u sustav dodajemo upravljačka mjesta tako da mjesta

$$\boxed{L m_p(k) + m_c(k) = b}$$

$$W = [W_p \; W_c] \qquad m = \begin{bmatrix} m_p \\ m_c \end{bmatrix}$$

$$W = \begin{array}{c|cccccccccccc}
 & PI & RA & RU1 & RU2 & RU3 & RS & RSA & S & SA & SZO & SZOA & PO \\
\hline
t_1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
t_2 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
t_3 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & 0 \\
t_4 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
t_5 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
t_6 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
t_7 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}$$

-1 (ulazi u prijelaz)

1 (izlazi iz prijelaza)

broj predmeta je ograničen na 5 ↓

$$\boxed{b = 5}$$

vektor $L = [0\ 0\ 0\ 0\ 0\ \underbrace{1\ 0\ 1\ 0\ 1}_{\text{na ovim mjestima želimo ograničiti broj oznaka}}\ 0\ 0]$

$m_p(0) \rightarrow$ vektor početnog stanja!

$$m_p(0) = [20\ 1\ 0\ 0\ 0\ 0\ 3\ 0\ 3\ 0\ 2\ 0]$$

$$W_c = -W_p \cdot L^T = -[0\ \ 1\ \ 0\ \ -1\ \ 1\ \ -1\ \ 0]^T$$

$$= [0\ \ -1\ \ 0\ \ 1\ \ -1\ \ 1\ \ 0]^T$$

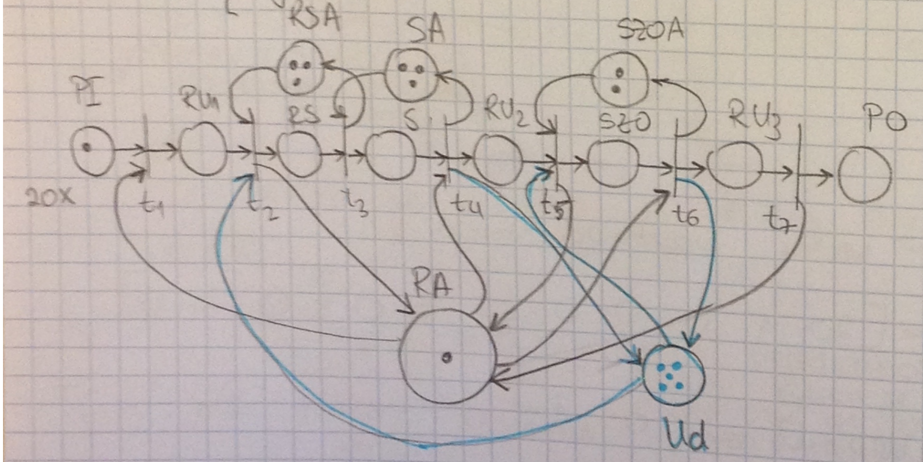
$$W_c = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix}$$

$$m_c(0) = b - L \cdot m_P^T(0) =$$

$$= 5 - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 20 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 3 \\ 0 \\ 3 \\ 0 \\ 2 \\ 0 \end{bmatrix} = 5 - 0 = 5$$

$\boxed{m_c(0) = 5}$ → broj oznaka upravlj. mjesta

$$W_c = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix}$$ → gdje treba upravlj. mjesto postaviti u mreži



- u ovom podzadatku je isti slučaj kao i pod c) samo što sada prioritet ima pražnjenje tj. upravljačko mjesto Ud_3 pa Ud_2 i onda Ud_1 s tim da sada imamo i dodatni upravljački algoritam zbog broja ograničenja predmeta

AKO (RS + S + SZO = = 5) ONDA ($Ud_1 = 0$, $Ud_2 = 0$, $Ud_3 = 1$)

2) 4 tipa resursa A, B, C i D i 5 procesa P, Q, R, S i T koji koriste te resurse

Tip resursa	A	B	C	D
Broj instanci	3	8	9	10

Stanje sustava:

$$
A = \begin{array}{c|c|c|c|c}
 & A & B & C & D \\
\hline
P & 0 & 0 & 1 & 2 \\
Q & 1 & 0 & 0 & 0 \\
R & 1 & 1 & 1 & 4 \\
S & 0 & 0 & 2 & 2 \\
T & 0 & 1 & 1 & 2
\end{array}
$$

$$
N = \begin{array}{c|c|c|c|c}
 & A & B & C & D \\
\hline
P & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\
Q & 2 & 7 & 5 & 0 \\
R & 1 & 2 & 5 & 2 \\
S & 0 & 6 & 2 & 0 \\
T & 3 & 2 & 8 & 4
\end{array}
$$

a) da li je trenutno stanje sigurno po Bankarevom algoritmu?

- zbroj vrijednosti u stupcu matrice A, te potom razlika vrijednosti broja resursi i dobivene vrijednosti iz matrice A definira vektor d -> br slobodnih resursa

$$d = \begin{bmatrix} 1 & 6 & 4 & 0 \end{bmatrix}$$

-> iz redaka matrice **N** možemo vidjeti postoji li proces koji se može završiti na način ako su svi elementi retka $\leq$ od elemenata vektora retka d

-> vidimo da se proces P može izvršiti ali isto tako i proces S. Oba retka u matrici N zadovoljavaju prethodni uvjet.

Odabiremo da se prvo može izvršiti proces P.

$$d = d + A_p(1,:) = \begin{bmatrix} 1 & 6 & 5 & 2 \end{bmatrix}$$

$$A = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 4 \\ \cancel{0} & \cancel{0} & \cancel{2} & \cancel{2} \\ 0 & 1 & 1 & 2 \end{bmatrix} \qquad N = \begin{array}{c} \\ \xrightarrow{1.)} P \\ Q \\ R \\ \rightarrow S \\ 2.)\ T \end{array} \begin{bmatrix} A & B & C & D \\ 0 & 0 & 0 & 0 \\ 2 & 7 & 5 & 0 \\ 1 & 2 & 5 & 2 \\ \cancel{0} & \cancel{6} & \cancel{2} & \cancel{0} \\ 3 & 2 & 8 & 4 \end{bmatrix}$$

-> u nastavku vidimo da proces S može da se završi

$$d = d + A_S(4,:) = \begin{bmatrix} 1 & 6 & 7 & 4 \end{bmatrix}$$

$$A = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 4 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 1 & 1 & 2 \end{bmatrix} \qquad N = \begin{array}{c} \\ P \\ Q \\ \rightarrow R \\ 3.)\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 2 & 7 & 5 & 0 \\ 1 & 2 & 5 & 2 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 3 & 2 & 8 & 4 \end{bmatrix}$$

→ sada se može izvršiti proces R jer u d vektoru ima dovoljno slobodnih resursa

vektor d postaje:

$$d = d + A_R(3,:) = [2\ \ 7\ \ 8\ \ 8]$$

$$A = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 1 & 0 & 0 & 0 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 1 & 1 & 2 \end{bmatrix} \qquad N = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 2 & 7 & 5 & 0 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 3 & 2 & 8 & 4 \end{bmatrix} \begin{array}{c} \\ \\ \leftarrow 4.) \\ \\ \\ \\ \end{array}$$

→ sada vidimo da može da se izvrši Q

$$d = d + A_Q(2,:) = [3\ 7\ 8\ 8]$$

$$A = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 1 & 1 & 2 \end{bmatrix} \qquad N = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 3 & 2 & 8 & 4 \end{bmatrix} \begin{array}{c} \\ \\ \\ \\ \\ \Leftarrow 5.) \end{array}$$

$$d = d + A_T(5,:) = [3\ \ 8\ \ 9\ \ 10] \ //$$

→ vektor d je istog iznosa kao i vektor ukupnog broja resursa što ukazuje na to da su svi procesi uspješno završeni

Početno stanje je sigurno prema Bankarovom algoritmu

Postoji više redoslijeda po kojem su se procesi mogli završavati, a ja sam radila redoslijed:

$$P \rightarrow S \rightarrow R \rightarrow Q \rightarrow T$$

b) ako proces T u zadanom stanju zahtjeva alokaciju resursa $\begin{pmatrix} A & B & C & D \\ 0 & 0 & 3 & 0 \end{pmatrix}$ da li mu treba odobriti zahtjev prema Bankarevom algoritmu?

$$\begin{pmatrix} A & B & C & D \\ 0 & 0 & 3 & 0 \end{pmatrix}$$

$$* \; A_{T_{poslije}} = -N_T + (A_{prije\,T} + N_{T_{prije}}) = -[0\,0\,3\,0] + [3\;3\;9\;6] = [3\;3\;6\;6]$$

$\rightarrow$

$$N = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ 0 & 0 & 0 & 0 \\ 2 & 7 & 5 & 0 \\ 1 & 2 & 5 & 2 \\ 0 & 6 & 2 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix} \quad // \quad A_2 = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 4 \\ 0 & 0 & 2 & 2 \\ 3 & 3 & 6 & 6 \end{bmatrix}$$

ako ovo stanje gledamo kao ukupan br resursa potrebnih procesu T minus broj potrebnih u matrici N *

$$d_0 = [1\;6\;4\;0]$$

$\rightarrow$ ako odradimo cijeli postupak kao u a) dijelu vidimo da se zahtjev $\boxed{\text{može}}$ odobriti, jer će se svi procesi izvršiti do kraja

//

(ili) ako gledamo da je matrica A jednaka početnom stanju (ali time se mijenja broj ukupnih resursa potrebni procesu T)

$$A_1 = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array} \begin{bmatrix} A & B & C & D \\ 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 4 \\ 0 & 0 & 2 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix} \rightarrow$$

sukladno opet se može dopustiti zahtjev!

(ili iz druge matrice A_2 (ne znam kako ste zamislili da riješimo)

$N_T \leq d$ ✓

$d_1 = d_0 + A_{1T} = [1\;7\;5\;2] \rightarrow N_P \leq d$

$d_2 = d_1 + A_1P = [1\;7\;6\;4] \rightarrow N_S \leq d$

$d_3 = d_2 + A_1S = [1\;7\;8\;6] \rightarrow N_R \leq d$

$d_4 = d_3 + A_R = [2\ 8\ 9\ 10] \to N_Q < d$

$d_5 = d_4 + A_Q = [3\ 8\ 9\ 10]$

$T \to P \to S \to R \to Q$

c) Neka je proizvoljan sustav, upravljan po Bankarevom algoritmu, u sigurnom stanju. Da li dodavanje novih procesa u sustav mijenja sigurnost stanja?

→ Sigurnost stanja ovisit će o dodanom procesu i ukupnom broju potrebnih resursa da se taj proces izvrši.
Ako sustav ima n procesa i ako je stanje sigurno, to znači da smo nakon n-iteracija dobili vektor d jednak vektoru r

$\boxed{d = r}$

- dodavanjem novih procesa kojima su za izvršavanje dovoljni dostupni (i postojeći) resursi sustava $\boxed{\text{ne mijenja se}}$

$\boxed{\text{Sigurnost stanja}}$ jer prilikom dodavanja novih procesa tim procesima nismo dodijelili niti jedan resurs što povlači da su retci matrice A za nove procese nul vektori, tj.:

$$A = \begin{array}{r} \\ n+1 \\ n+2 \\ n+N \end{array} \begin{bmatrix} & A \text{ od prije} & & \\ 0 & 0 & 0 & 0 \ \cdots \\ 0 & 0 & 0 & 0 \ \cdots \\ 0 & 0 & 0 & 0 \ \cdots \end{bmatrix}$$

$$N_{\substack{n+1 \\ \vdots \\ n+N}} \le d = r$$

→ kao što je spomenuto, ako je za dodane procese broj potrebnih resursa za njihovo izvršavanje veći od dostupnih resursa koje sustav ima u vektoru r onda će stanje biti nesigurno jer se dodani proces ne može izvršiti.

$N_{n+1} > d = r$