

ALGORITMI I STRUKTURE PODATAKA I



Pripremila: Lena Joković

KB-23

1. a) - неустерен, неминимален граф, само са позитивни тежестина
- ивовови моделују буђе, а тине прираће ње
 - прираће ње тине су међусобно повезане компоненте графа, иј - сутовн међусобно достижнине ивовов

8) GROUPS(Friendships, k, n)

$V = \emptyset, E = \emptyset$

for $\ell = 1$ to k do

$i, j, \text{quality} = \text{Friendships}[\ell]$

if $\text{quality} > 0$ then

$V = V \cup \{i, j\}$

$E = E \cup \{(i, j)\}$

$\text{visit}[i] = \text{visit}[j] = \text{false}$

end-if

end-for

$\text{num} = 0$

for each $i \in V$ do

if (not($\text{visit}[i]$)) then

$\text{num} += 1$

$\text{BFS}(V, E, \text{groups}, \text{num}, i)$

end-if

end-for

return groups.

BFS($V, E, \text{groups}, \text{num}, i$)

$\text{visit}[i] = \text{true}$

$\text{groups}[\text{num}] = \{i\}$

INSERT(Q, i)

while (not(QueueEmpty(Q))) do

$v = \text{DELETE}(Q)$

for $\{w : (v, w) \in E\}$ do

if (not($\text{visit}[w]$)) then

$\text{visit}[w] = \text{true}$

$\text{groups}[\text{num}] += w$

INSERT(Q, w)

end-if

end-for

end-while

KB-23

2) а) - мотулу тогловулу тогровулу:

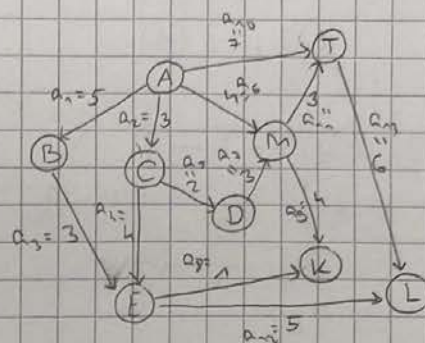
ABCEDMKTL

ACBEDMKTL

ABCEDEMKTL

ABCEDMTKL

ABCEDMTLKL



б)

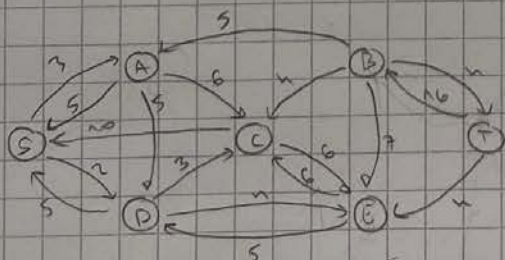
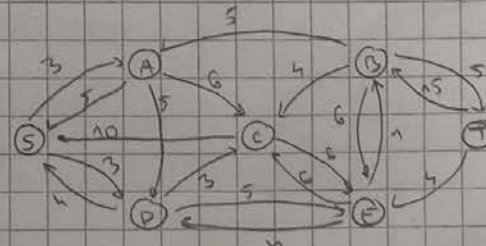
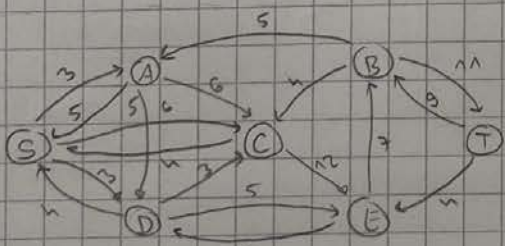
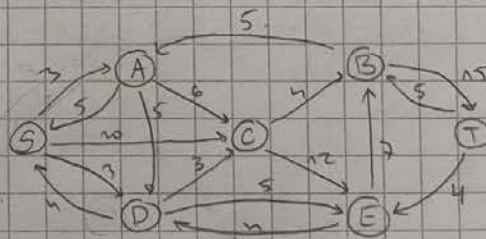
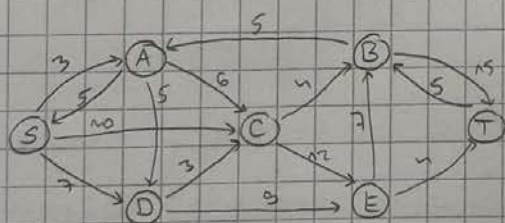
V	EST	LST	L
A	0	0	0
B	5	9	4
C	3	3	0
E	8	12	4
D	5	5	0
M	8	8	0
K	12	17	5
T	11	11	0
L	17	17	0

E	Q
a ₁	4
a ₂	0
a ₃	4
a ₄	5
a ₅	0
a ₆	4
a ₇	0
a ₈	8
a ₉	5
a ₁₀	4
a ₁₁	0
a ₁₂	4
a ₁₃	0

⇒ критичант пуги:

ACDMTL

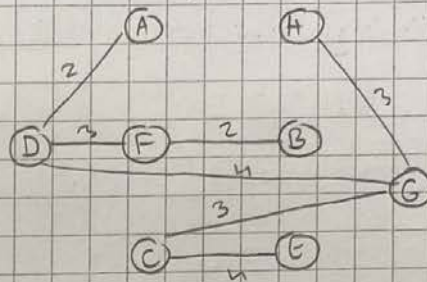
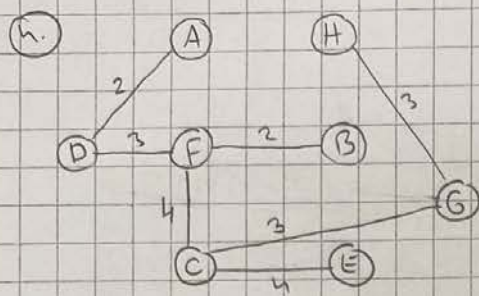
3.



⇒ максималн пропуск: 20

- ние јединствено решение

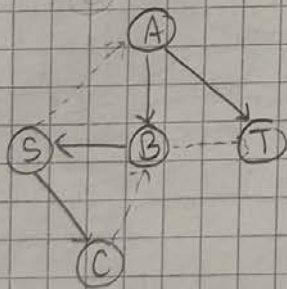
K3-23



- додавање грана минималне тежине док цео граф није повезан, без прављења циклуса
- два различита минимална обједињена $\Rightarrow$ два

- 5) - постојање циклуса $\Leftrightarrow$ повратна грана у DFS
 - повратна грана у BFS

- грана је повратна ако води до већ посећеног чвора и ако је тај чвор претходник посматраног чвора у истом области



- ABS је циклус јер је SA повратна грана
- ABT није циклус јер је BT повратна грана

K3-23

(6.) a) - матрица следована T : $T[i, j] = 0$ ако је $i = j$ или $W[i, j] = \infty$ $T[i, j] = j$ ако је $i \neq j$ и $W[i, j] < \infty$ - првимом позитивном пераукацијом пута $i-j$ преко k : $T[i, j] = T[i, k]$ 8) PATH(T, i, j) $k = T[i, j]$ if $k = 0$ then

print("Nema puta")

return

end-if

print(i)while($k \neq 0$) doprint(k) $k = T[k, j]$

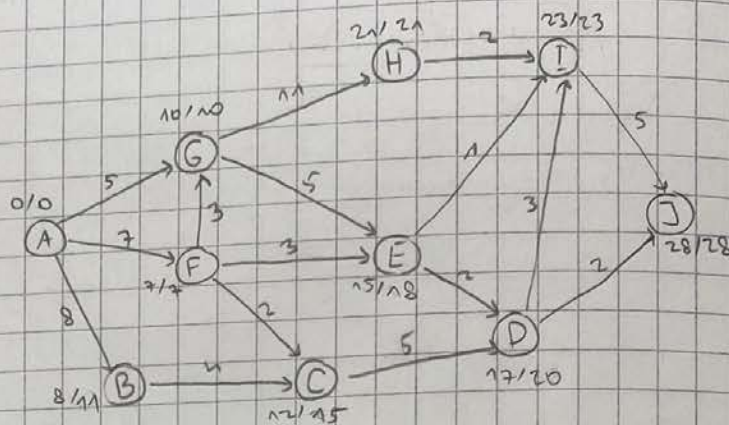
end-while

mas da ovo printa u
obrnutom redosledu
proveri!

КЗ-22

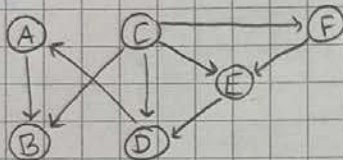
1.

чвор	EST	LST	L
A	0	0	0
B	8	11	3
F	7	7	0
G	10	10	0
C	12	15	3
E	15	18	3
H	21	21	0
D	17	20	3
I	23	23	0
J	28	28	0



⇒ критичан пут: A-F-G-H-I-J

2.



P:

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	0	0	0	0
C	0	1	0	1	1	1
D	1	0	0	0	0	0
E	0	0	0	1	0	0
F	0	0	0	0	1	0

P^A:

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	0	0	0	0
C	0	1	0	1	1	1
D	1	1	0	0	0	0
E	0	0	0	1	0	0
F	0	0	0	0	1	0

P^F:

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	0	0	0	0
C	1	1	0	1	1	1
D	1	1	0	0	0	0
E	1	1	0	1	0	0
F	0	0	0	0	1	0

P^F:

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	0	0	0	0
C	1	1	0	1	1	1
D	1	1	0	0	0	0
E	1	1	0	1	0	0
F	1	1	0	1	1	0

- матрица достижимости

- Граф је унилатерал, јер на свакој
директној линији има 1, тј. из једног чвора
може доћи до сваког чвора од њега самог

123-22

③ HWE($G, \text{start}, \text{end}, k$) $U = \{\text{start}\}$ $E' = \emptyset$ $p = 0$ while ($\text{end} \notin U$) do $\text{min} = \infty$ for each $i \in U$ do for $j = 0$ to n do if $w[i, j] < \text{min}$ and $j \notin U$ then $\text{min} = w[i, j]$ $U = i, v = j$

end-if

end-for

end-for

 $U = U + \{v\}$ $E' = E' + \{(u, v)\}$ $p = \text{min}$

end-while

if $p > k$ then

return false

else

return true

end-if

K3-22

(h) CALC-SHORTEST-PATH($G, n, \text{startCity}, \text{middleCity}, \text{endCity}$)

$S = \{\text{startCity}\}$

for $i = 1$ to n and $i \neq \text{startCity}$ and $i \neq \text{endCity}$ do

$d[i] = w[\text{startCity}, i]$

if $w[\text{startCity}, i] \neq \infty$ then

$t[i] = \text{startCity}$

else $t[i] = 0$

end-if

end-for

while ($\text{middleCity} \notin (V-S)$) do

find $\min \{d[i] : i \in (V-S) \text{ and } i \neq \text{endCity}\}$

$S = S + \{i\}$

for each $j \in (V-S)$ do

if $(d[i] + w[i, j] < d[j])$ then

$d[j] = d[i] + w[i, j]$, $t[j] = i$

end-if

end-for

end-while

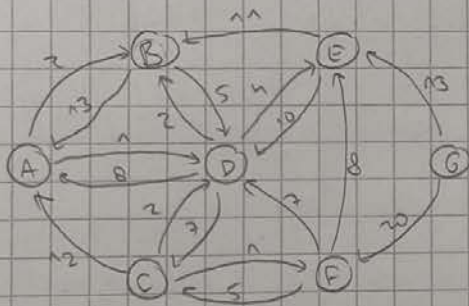
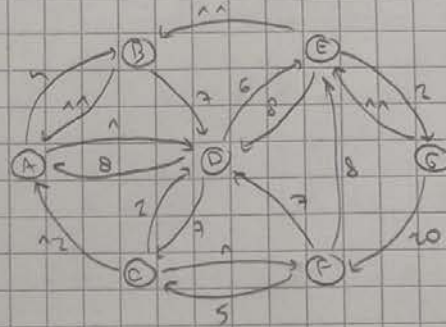
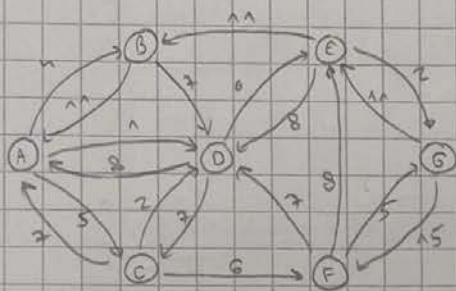
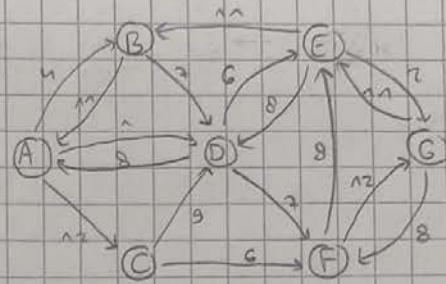
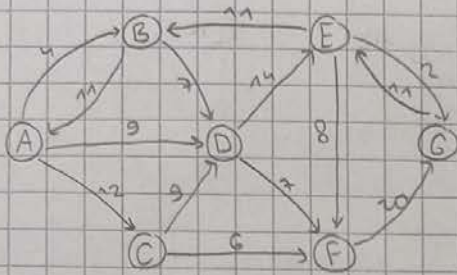
WRITE-PATH($t, \text{startCity}, \text{middleCity}$)

($w_{mid} \sim 3$ or middleCity go endCity)

WRITE-PATH($t, \text{middleCity}, \text{endCity}$)

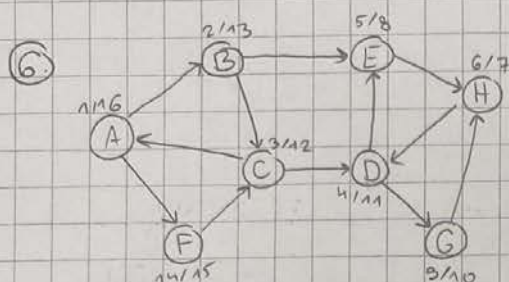
123-22

5.



=> максимальный поток: 33

K3-22

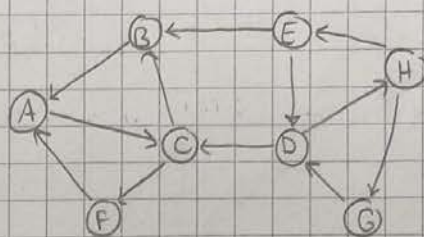


- DFS поопейна и завршна
времена обласна G

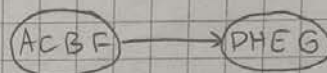
$\Rightarrow$ DFS за Gt за A и D

$\Rightarrow$ јавно повезане компоненти

ACBF и DHEG



- транзитиван граф Gt



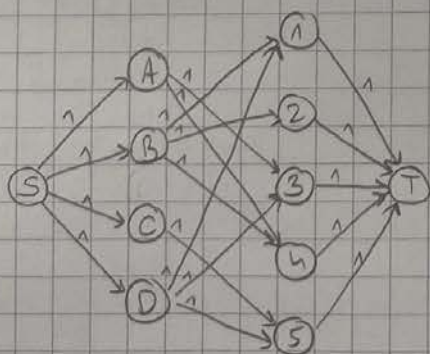
- резуцирван граф

⑦ - додавање нбора S повезанот со једном групом нборова и
нбора T повезанот со другом групом нборова

- устерикување свих трана од S до једне групе, од једне групе
до друге, и од друге групе до T

- доделување ланој трани тежину 1

- примера Форд-Фулмерсоновот алгоритам



K3-22

- 8) a) - ширинско обухватно стадо је обухватно стадо
 добијено BFS обилазном графа
 - сваки пут када се три обилазну дође до непосећеног
 чвора, та долазна трага се дога у обухватно стадо

8) FIND-TRANSVERSE-EDGES(G, n, x)

```

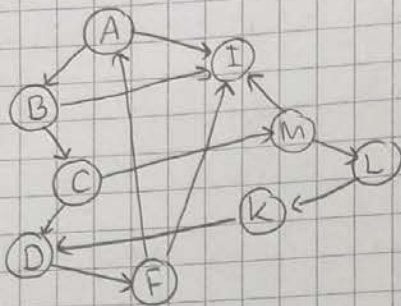
for  $i = 1$  to  $n$  do
    visit[i] = false
    parent[i] = 0
end-for
visit[x] = true
INSERT(Q, x)
S =  $\emptyset$ 
while (not (QUEUE-EMPTY(Q))) do
    v = DELETE(Q)
    for  $\{w : (v, w) \in E\}$  do
        if (not (visit[w])) then
            visit[w] = true
            parent[w] = v
            INSERT(Q, w)
        else if (parent[w]  $\neq$  v) then
            A = A + (v, w)
        end-if
    end-for
end-while
return A

```

ACD

K3-19

1.



BFS:

ABICDMFLK,

DFS:

AIBCMFLKDF

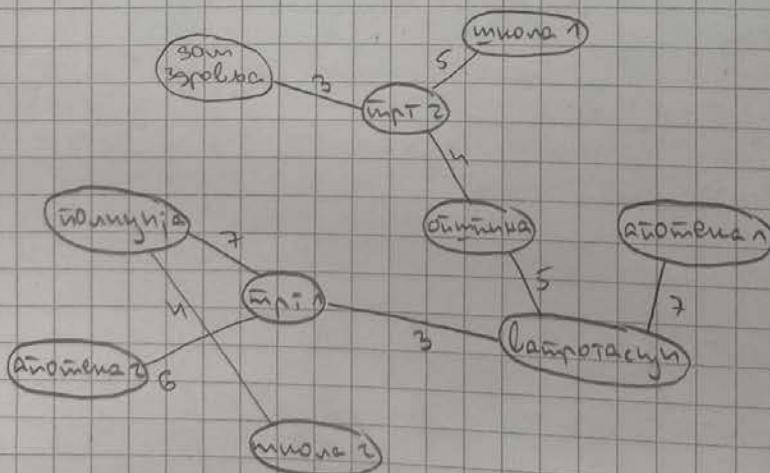
a) BFS: QUEUE

A
B I
 I C
 C
 DM
 MF
 F L
 L
 K

б) DFS: STACK

A
B I
B
C
DM
DL
DK
DD
D
F

2. - Прилож или Крускалов алгоритъм
⇒ минимално обхващащо съдържание



- добавяне на най-малките тежесте, без образуване на цикъл

КЗ-13

3. а) - укажени граф, чворови представљају инструкције,
а гране зависности између изразних операнда

б) - може се користити и TOP SORT

б) ADD A, B, C

MUL C, E, D

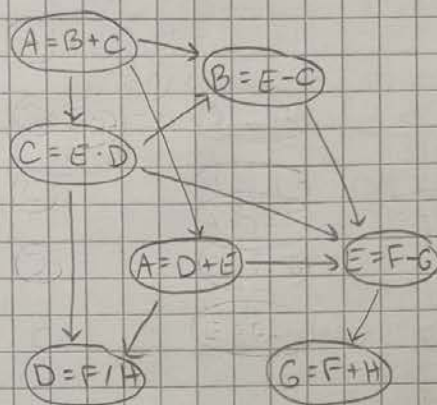
SUB B, E, C

ADD A, D, E

SUB E, F, G

DIV D, F, H

ADD G, F, H



ADD A, B, C

ADD A, D, E

MUL C, E, D

SUB B, E, C

DIV D, F, H

SUB E, F, G

ADD G, F, H

н) K-DISTANCE(G, n, k, id)

QUEUE-INIT(Q)

INSERT(Q, id)

INSERT(Q, nil)

while (k > 0) do

 v = DELETE(Q)

 if v = nil then

 k = k - 1

 else

 for {w: (v, w) ∈ E} do

 INSERT(Q, w)

 end-for

 end-if

end-while

while ((v = DELETE(Q)) ≠ nil) do

 print(v)

end-while

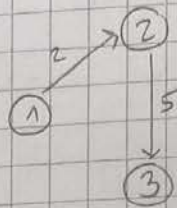
ACD

V3-19

5. a)

$$D = \begin{bmatrix} 0 & 2 & 7 \\ \infty & 0 & 5 \\ \infty & \infty & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$



6) PATH(T, A, B, C)

STACK-INIT(S)

t = C

while (t ≠ 0) do

 PUSH(S, t)

 t = T[B, t]

end-while

t = B

while (t ≠ A) do

 t = T[A, t]

 PUSH(S, t)

end-while

while not (STACK-EMPTY(S)) do

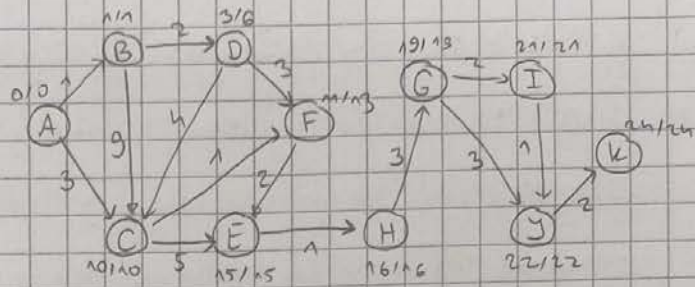
 print(POP(S))

end-while

K3-13

6.

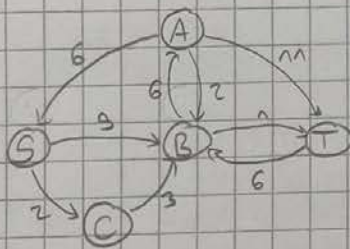
узор	EST	LST	L
A	0	0	0
B	1	1	0
D	3	6	3
C	10	10	0
F	11	13	2
E	15	15	0
H	16	16	0
G	19	19	0
I	21	21	0
J	22	22	0
K	24	24	0



=> тојолошки поредок: A B C F E H G I J K

критични пати: A B C E H G I J K

7. - два карактеристична начина за избор пати доветанот
 проток: пати максималниот резидуалниот капацитет
 и пати са минималниот бројем транса



максимални капацитет: SBAT

минимални бр. транса: SBT

КЗ-19)

(8) а) - сложность Дijkstra-ного алгоритма:

 $O(n^2)$ за матричную репрезентацию $O(e + n \log n)$ за список смежности

- зависит от способа выбора вершины минимального расстояния, и от способа прохода через все смежные вершины при релаксации

б) DIJKSTRADJLIST(G, x, y) $S = \{x\}$ temp = $G[x] \rightarrow next$ while (temp $\neq$ nil) do $d[temp.info] = temp.w$ $t[temp.info] = x$ temp = temp $\rightarrow$ next

end-while

while (y $\notin$ S) do

?

K3-18)

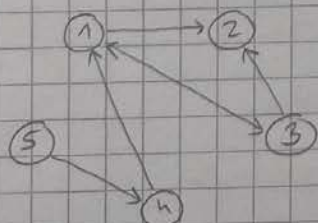
- ① - за сортиране графа у инверзном топологичном поретку, гранично
 ниворе без излазних трана, уклањамо их и постављамо докучили
 - листе суседности: нвор без излазних трана налазило трансном
 кроз затлавање, а уклањамо та трансном кроз све листе
 - инверзне листе суседности: нвор без излазних трана налазило
 трансном кроз све листе, а уклањамо та директно

?

$$A^{(1)} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad A^{(2)} = A^{(1)} \cdot A^{(1)} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A^{(3)} = A^{(2)} \cdot A^{(1)} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\Rightarrow A^{(1,2,3)} = A^{(1)} + A^{(2)} + A^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$



V3-18

③ FIND_PATHS_NUM(G, src, dst, k)

$T = \text{TOP_SORT}(G)$

$isrc = \text{FIND}(T, src)$

$idst = \text{FIND}(T, dst)$

$num = \text{INIT_MATRIX}(n, k+1)$

$num[isrc][0] = 1$

for $i = isrc$ to $idst - 1$ do

$node = T[i]$

 for $\{v : (node, v) \in E\}$ do

 for $j = 0$ to k do

 if $j + w(node, v) \leq k$ then

$num[v][j + w(node, v)] += num[node][j]$

 end-if

 end-for

 end-for

end-for

$sum = 0$

for $i = 1$ to k do

$sum += num[idst][i]$

end-for

return sum

FIND(T, x)

for $i = 1$ to n do

 if $T[i] = x$ then

 return i

 end-if

end-for

return -1

123-18

④ TELECOM(G, n, k)

$E' = \emptyset$

for each $(u, v) \in E$ do

 PQ-INSERT(PQ, $w(u, v)$)

end-for

num = 0

while (num < n-1) do

$w(u, v) = \text{PQ-MIN-DELETE}(PQ)$

 if $(w(u, v) > k)$ then

 return false

 end-if

 if $((u \in T_i) \text{ and } (v \in T_j) \text{ and } (i \neq j))$ then

$E' = E' \cup \{(u, v)\}$

$T_{i \cup j} = T_i \cup T_j$

 num += 1

 end-if

end-while

return true

K3-18

⑤ CONN-COMP-MAT(G)

num = 0

a = INIT-ARRAY(n)

for i = 1 to n do

visit[i] = false

end-for

for i = 1 to n do

if (not(visit[i])) then

num = num + 1

a[num] = i

BFS-VISIT(G, i)

end-if

end-for

return a, num

BFS-VISIT(G, i)

visit[i] = true

Q = QUEUE-INIT(n)

INSERT(Q, i)

while (not(QUEUE-EMPTY(Q))) do

i = DELETE(Q)

for j = 1 to n do

if (G[i][j] and not(visit[j])) then

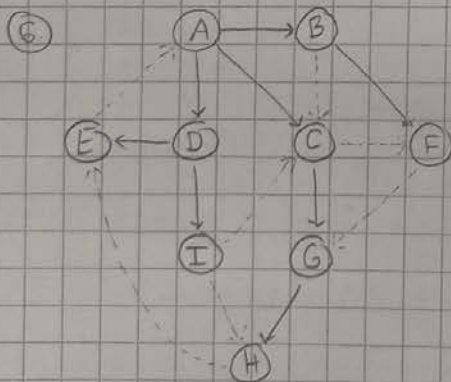
visit[j] = true

INSERT(Q, j)

end-if

end-for

end-while



BFS от A

Грaне сmаdнa:

AB, AC, AD, BF, CG, DE, DI, GH

Грaне yнoтpег:

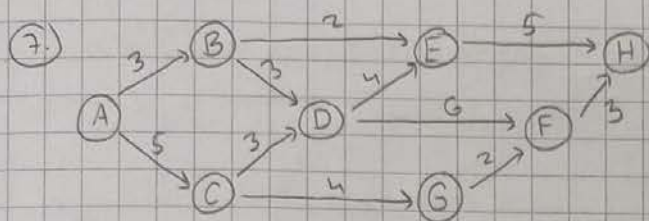
Пoвpащeннe гpaнe:

EA

Пoтpягнeннe гpaнe:

BC, CF, FG, IC, IH, HE

K3-18



S	min nbor	A	B	D	E	F	G	H
C	-	∞	∞	3	∞	∞	4	∞
C, D	D	∞	∞	3	7	9	4	∞
C, D, G	G	∞	∞	3	7	6	4	∞
C, D, G, F	F	∞	∞	3	7	6	5	9
C, D, G, F, E	E	∞	∞	3	7	6	5	9
C, D, G, F, E, H	H	∞	∞	3	7	6	5	9

- алгоритъм завъртава ред нареден б. корана, каде је најмање растојање до необрађеног чвора ∞

8. а) $GF(V, EF)$, $EF = \{(u, v) : u, v \in V, cf(u, v) > 0\}$

$cf(u, v) = c(u, v) - f(u, v)$ - резулант напона

$f(u, v) \leq c(u, v)$, $f(u, v) = -f(v, u)$, $\forall (u, v) \in E$

- резулант грџ може бити максимум 2 е трага

б) G to GF(G)

$GF = \text{MALLOC}(n)$

for $i = 1$ to n do

temp = $G[i] \rightarrow \text{next}$

while (temp) do

if (temp.c - temp.f > 0) then

APPEND($GF[i]$, temp)

end-if

temp = temp $\rightarrow$ next

end-while

end-for return GF

APPEND($GF[i]$, node)

new.info = node.info

new.cf = node.c - node.f

temp = $GF[i]$

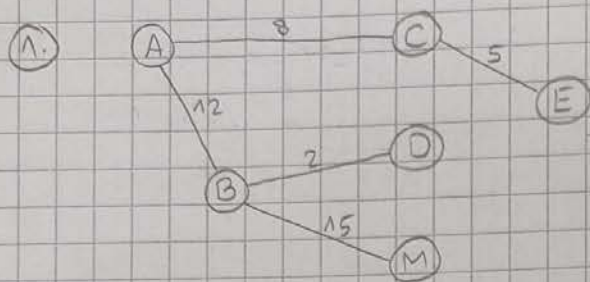
while (temp $\rightarrow$ next) do

temp = temp $\rightarrow$ next

end-while

temp $\rightarrow$ next = new

K3-17



- догадване, че на минимална
 стойност докато цената е еднаква
 няма по-добро, без да се търси
 цялостно

2. TOPSORT(G, n)

 $G' = G$

 for $i = 1$ to n do

 $temp = G'[i] \rightarrow next$

 while ($temp$) do

 $in[temp.info] += 1$
 $temp = temp \rightarrow next$

end-while

end-for

 for $i = 1$ to n do

 $num = -1$

 for $j = 1$ to n do

 if ($in[j] = 0$ and $G[j].num > num$) then

 $G[j].num = num$
 $v = j, num = G[j].num$

end-if

end-for

 $T[i] = v$
 $G'[v].num = -1$
 $temp = G'[v] \rightarrow next$

 while ($temp$)

 $in[temp.info] -= 1$
 $temp = temp \rightarrow next$

end-while

end-for

 return T

КЗ-17

- ③ а) - Нечислені метричні граф, вловлюючи тосао, мити
 в дві распрямку на пугу, граф уннуе, а метрике
 граф спої мити на мити уннуе

δ) GET-PERICA-HOME(G, work, home)

$S = \{\text{work}\}$

$i = \text{home}$

for ($i = 1$ to n) and ($i \neq \text{work}$) do

while ($t[i] \neq 0$) do

$d[i] = w[\text{work}, i]$

PUSH(S, i)

if ($w[\text{work}, i] \neq \infty$) then

$i = t[i]$

$t[i] = \text{work}$

end-while

else $t[i] = 0$

while (not(STACK-EMPTY(S))) do

end-if

$i = \text{POP}(S)$

end-for

print(i)

while ($\text{home} \in (V-S)$) do

end-while

find min $\{d[i] : i \in (V-S)\}$

$S = S + \{i\}$

for each $j \in (V-S)$ do

if ($d[i] + w[i, j] < d[j]$) do

$d[j] = d[i] + w[i, j]$

$t[j] = i$

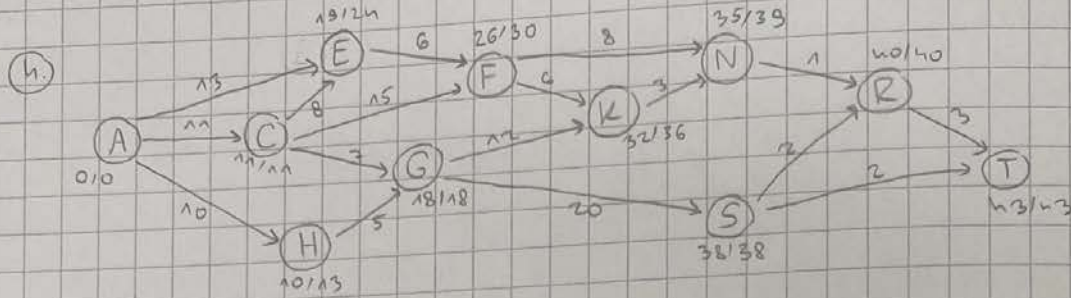
end-if

end-for

end-while

Acn

KB-17



Актив	EST	LST	L
A	0	0	0
C	11	11	0
E	19	24	5
H	10	13	3
F	26	30	4
G	18	18	0
K	32	36	4
N	35	39	4
S	38	38	0
R	40	40	0
T	43	43	0

=> критический путь:

ACGSRT

⑤ ?

KB-17

ACD

⑥ MAX-STRONG-CC(G, n)

$G_t = \text{INIT_MATRIX}(n, n)$

for $i=1$ to n do

for $j=1$ to n do

$G_t[i][j] = G[j][i]$

end-for

$\text{visit}[i] = \text{false}$

end-for

$\text{max} = 0$

for $i=n$ down to 1 do

if (not($\text{visit}[i]$)) then

$a, \text{num} = \text{DFS}(G_t, n, i, 0)$

if ($\text{num} > \text{max}$) then

$\text{max} = \text{num}$

$a_{\text{max}} = a$

end-if

end-if

end-for

return $a_{\text{max}}, \text{max}$

DFS(G, n, i, num)

$\text{visit}[i] = \text{true}$

$a[\text{num}++] = i$

for $j=1$ to n do if ($\text{visit}[j]$) then

if ($G[i][j]$ and (not($\text{visit}[j]$))) then

$\text{DFS}(G, n, j, \text{num})$

end-if

end-for

return a, num

ACM

КЗ-17

7. a) FLOW-CAP(F, R)

$C = \text{INIT_MATRIX}(n, n)$

$e = 0$

for $i = 1$ to n do

for $j = 1$ to n do

$C[i][j] = R[i][j] + F[i][j]$

if $(C[i][j] > 0)$ then

$e = e + 1$

end-if

end-for

end-for

return C, e

б) - резултатът трябва може да има наймалко е трана, а
найвече 2 е трана, ако је е број трана почетног
проточног графа

КЗ-17)

- 8) а) - повезан неусмерени граф је ацикличан само ако има $n-1$ тачку, па је потребно пребројати тачке у графу
 - временска сложеност је $O(n)$, где је n број чворова,
 јер је граф са n чворова и бар n тачака сигурно ацикличан

б) IS_CYCLIC(G)

for $i=1$ to $n-1$ do

find $\{(u,v) \in E\}$

$E = E - (u,v)$

end-for

if $(E = \emptyset)$

return false

else

return true

end-if