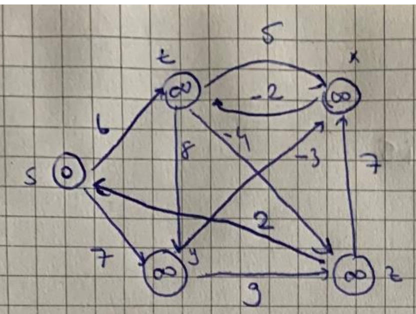
1-)



1. relax (t,x):

d[x] = ∞   d[x] > d[u] + 5   , nothing change

d[t] = ∞

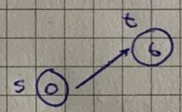
Basically nothing changes until we get to the source vertex.

2. relax (s,t)

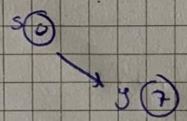
d[s] = 0   d[t] > d[s] + 6

d[t] = ∞

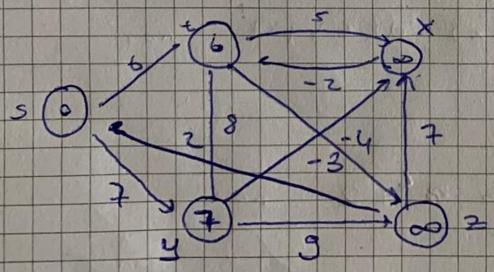
so d[t] = 0+6



3. relax (s,y)



Afer first iteration:



2nd iteration

1. relax (t, x)

$d[t] = 6$ $\quad$ $d[x] = 6 + 5$

$d[x] = \infty$ $\quad$ $\underline{\underline{d[x]}} = 11$

2. relax (t, y)

$d[t] = 6$ $\quad$ $d[y] > 6 + \infty$ no x

$d[y] = 7$

3. relax (t, z)

$d[t] = 6$ $\quad$ $d[z] = 6 + -4$

$d[z] = \infty$ $\quad$ $\underline{\underline{d[z]}} = 2$

4. relax (x, t)

$d[x] = 11$ $\quad$ $d[t] > d[x] + -2$, no

$d[t] = 6$

5. relax (y, x)

$d[y] = 7$ $\quad$ $d[x] > 7 + -3$, yes

$d[x] = 11$ $\quad$ $\underline{\underline{d[x]}} = 4$

6. relax (y, z)

$d[y] = 7$ $\quad$ $d[z] > 7 + 9$ no.

$d[z] = 2$

7. relax (z, x)

$d[z] = 2$ $\quad$ $d[x] > d[z] + 7$ no,

$d[x] = 4$

8. relax (z, s)

$d[s] = 0$ $\quad$ $0 > 0^{2} + 2$ no.

$d[z] = 2$

9. relax (s, t)

$d[t] = 6$ $\quad$ $6 > 0 + 6$ no

$d[s] = 0$

10. same as above.

4th iteration:

1. relax (t, x)

d[t] = 2 4 > 2+5 no
d[x] = 4

2. relax (t, y)

d[t] = 2 7 > 2+8 no,
d[y] = 7

3. relax (t, z)

d[t] = 2 2 > 2 + -4 yes
d[z] = 2

d[z] = -2

4. relax (x, t)

d[t] = 2 no, 2 > 4 + -2
d[x] = 4

5. relax (y, x)

d[y] = 7 4 > 7 + -3 no,
d[x] = 4

6. relax (y, z)

d[y] = 7 -2 > 7 + 16 no.
d[z] = -2

7. relax (z, x)

d[x] = 4 4 > -2 + 7 no,
d[z] = -2

8. relax (z, s)

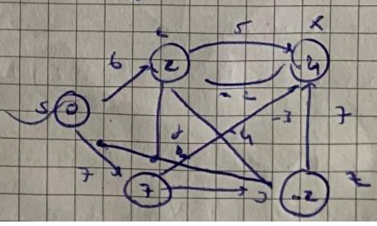
d[s] = 0 0 > -2 + 0 no,
d[z] = -2

9. relax (s, t) =

d[s] = 0 no,
d[t] = 2

10. same as above.

Graph:



5th iteration:

1. relax (t, x):

d[t] = 2   no change.
d[x] = 4

2. relax (t, y)   no change.

3. relax (t, z):

d[z] = -2   no change
d[t] = 2

4. relax (x, t)   no change

5. relax (y, x)   no change

6. relax (y, z)

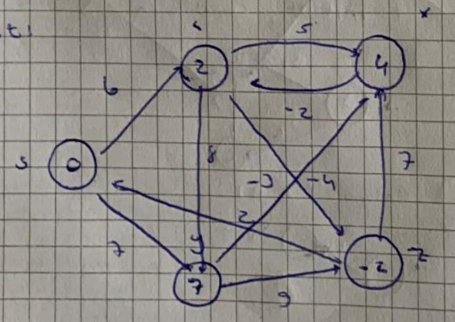
d[z] = -2   no change
d[y] = 7

7. relax (z, x)

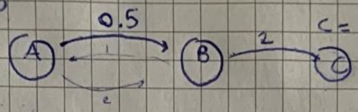
d[x] = 4   -2 + 7 = 5   no change
d[z] = -2

8, 9, 10 → no change.

Last:



10



C = S amount of A

= S. c(A,B) in B

c(A,B) is defined iff A,B has an edge

c(A,B) > 0

a-) S amount of A (10) c(A,B) = 0.5

S B = 5

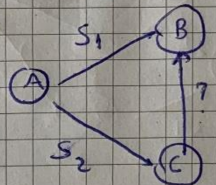
c = 10, therefore it is chained.

$\prod_{i=1}^{k-1} c(a_i, a_{i+1})$ ← the exchange rate is the exchange rate of intermediate rates, multiplied by the proceeding rates.

b-) ~~If exchanging from A to B is~~

$\frac{c(A,B)}{c(B,A)} > 1$, so there should be a loop where exchange rate is bigger than 1.

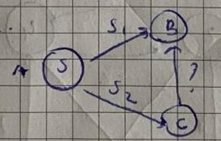
c-)



initial node (A)

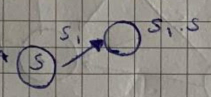
initial condition S

1.



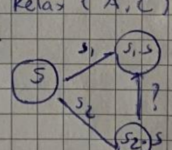
Relax (A,B)

2.



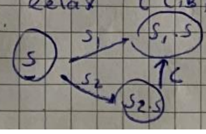
Relax (A,C)

3.



relax (C,B)

4.



if $c.S_2.S > S_1.S$

$V[B] = c.S_2.S$

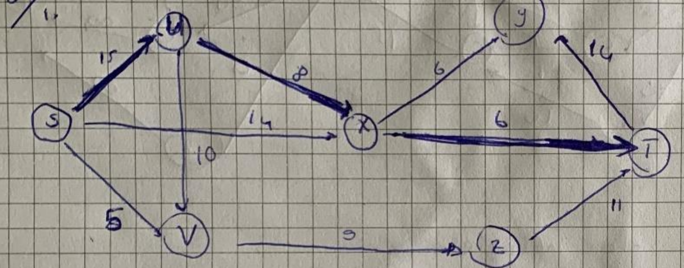
$\pi[B] = C$.

d. By multiplying the edge values with the d[v], we can use the bellman-Ford algorith to compute ~~shortest~~ the most optimal sequence. If there exist a loop, where we can get infinitely rich, we can notice it by using this algorithm.

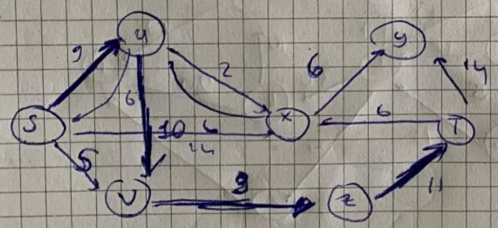
e. No, because Dijkstra can not find the ~~ultimate~~ infinitely rich sequence.

Here y is the sink.

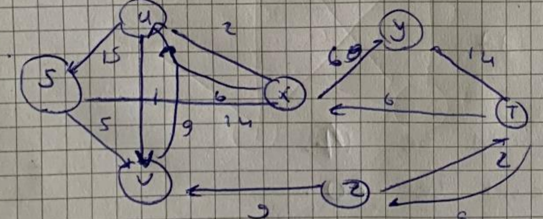
3./ 1.

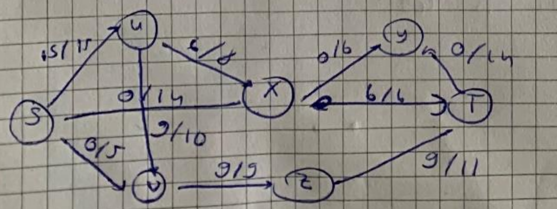


2.



3.



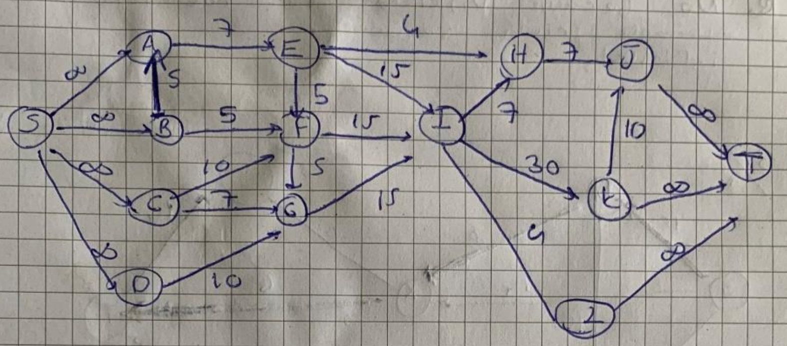


15

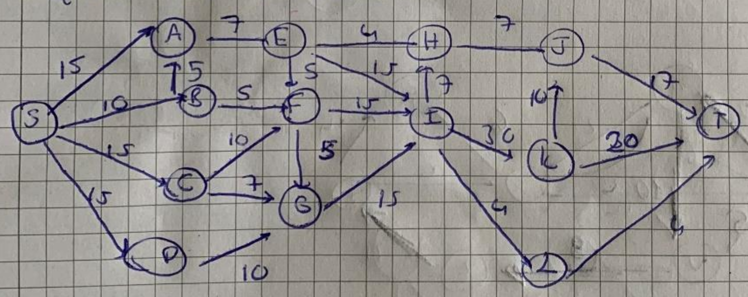
4/

a-) The edges represents the capacities we need to find the maximum flow to J, K and L.

1. Introduce a dummy sink and dummy source.



- Replace the infinities.

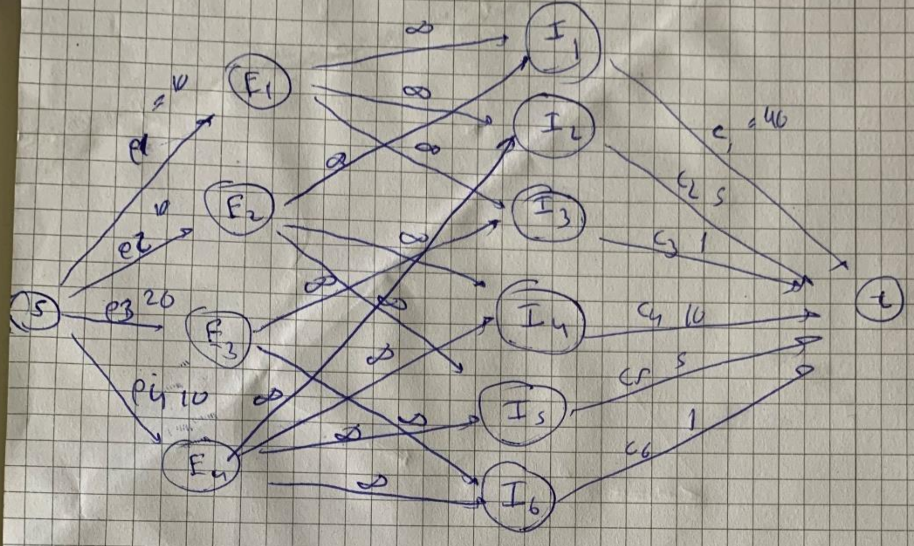


Max flow is 37.

b-) (I,L) should be 14.
(A,E) should be 20.

But max flow is 49.

5.



· We need to compute minimum cut s-t : (A,B)

↳ the projects in {A}-{s} is the set of projects that maximizes the profit.

profit = $-\text{Cost}(A) + \text{benefit}(A)$

$\text{cost}(A) = \sum_{c_j \in A} c_j$

~~pro~~ $\text{benefit}(A) = \sum_{p_j \in A} p_j$