Week11

Exercise 1: Simulate Dijkstra's algorithm to find shortest paths starting from node a in the given graph. Assume that the adjacency lists are in alphabetical order. In your simulation, show the intermediate results of the computation (lengths of the found paths to each node and the predecessor nodes) after each step.

<u>Solution:</u>

The adjacency lists in this graph G(V,E), where V ϵ {a, b, c, d, e, f} are in alphabetical order:

a – b- c – d
b – c
c - ∅
d – b

The source node s = a.

Dijkstra's algorithm:

Fig.1: Initialize-Single-Source(G,a)

for each vertex v ϵ G.v
v.d = ∞
v.p = NIL
a.d = 0
a.p = NIL

S = 0
while {a, b, c, d, e, f} ∉ S
choose a ϵ V \ S,
u = a
for all v = {b,f} ϵ vierus[a]
Relax (a,v,w)
b.d = 0 + 20 = 20, 20 < ∞
f.d = 0+2 = 2, 2 < ∞

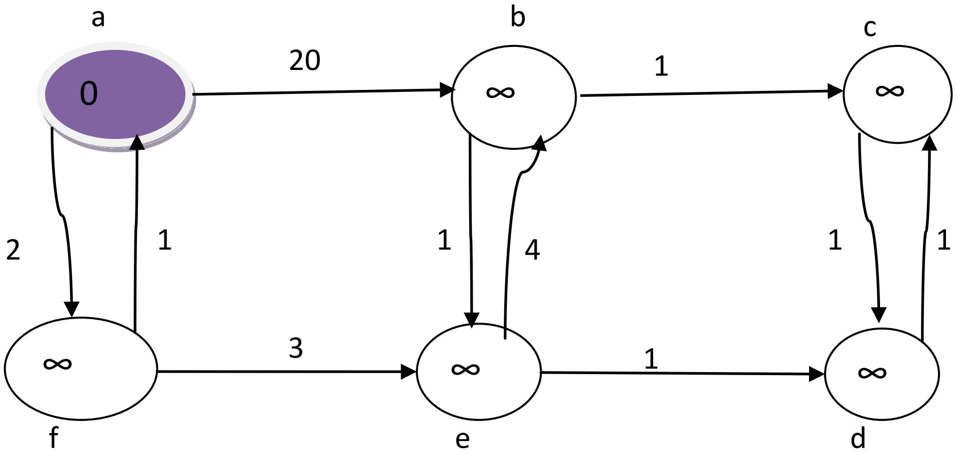


Fig.2: S = S ⋃ a

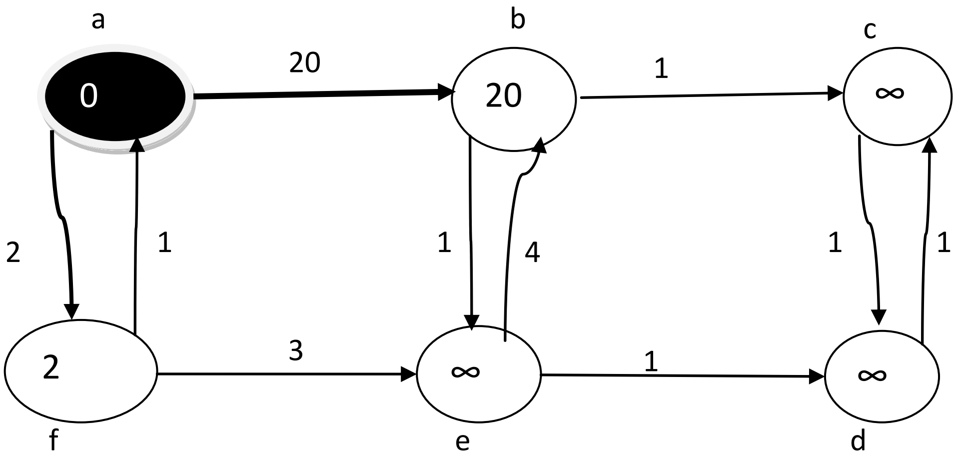


Fig.3: choose f ϵ V \ S

while {b, c, d, e, f} ∉ S
choose f ϵ V \ S,
u = f //since d.f = min {d.f, d.b} = min{2,20}
for all v = {e} ϵ vierus[f]
Relax (f,e,w)
e.d = 2 +3 = 5, 5 < ∞
e.p = f

f.p = a

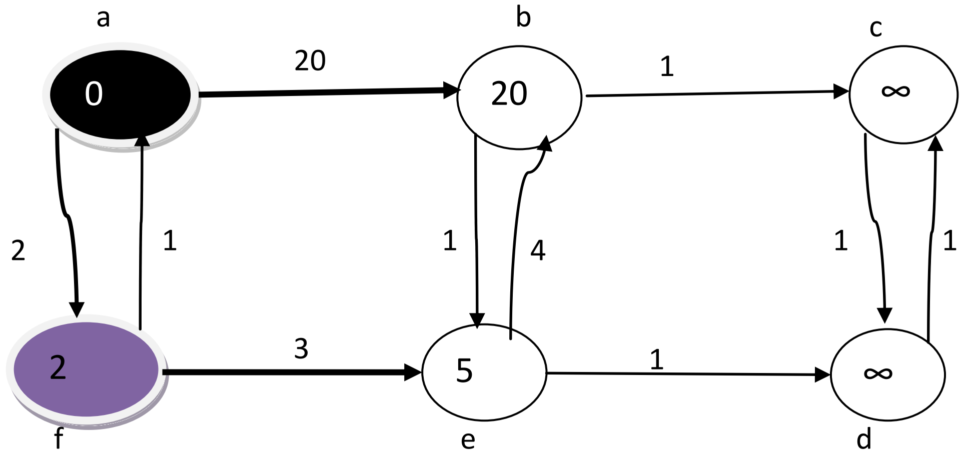


Fig.4: S = S ∪ *f*

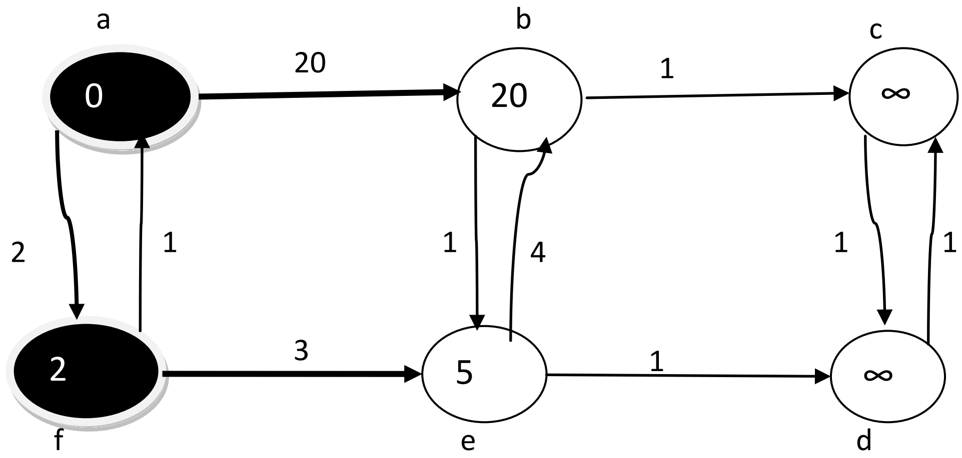


Fig.5: choose e ϵ V \ S

while {b, c, d,e} ∉ S

choose e ϵ V \ S,

u = e //because d.e = min{d.b, d.e} = min{20,5}

for all v = {b,d} ϵ vierus[e]

Relax (e,v,w)

d.d = 5 +1 = 6, 6 < ∞

b.d = 5 +4 = 9, 9 < 20

d.p = e
b.p = e

e.p = f

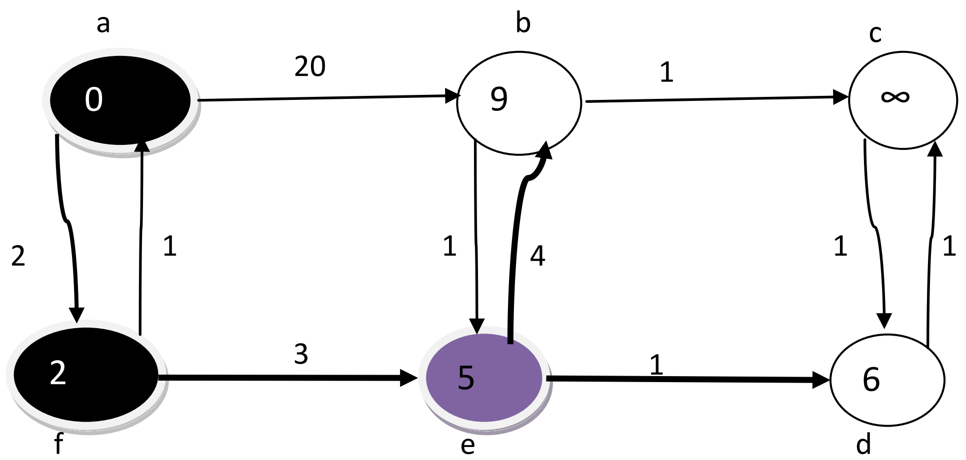


Fig.6: S = S ∪ e

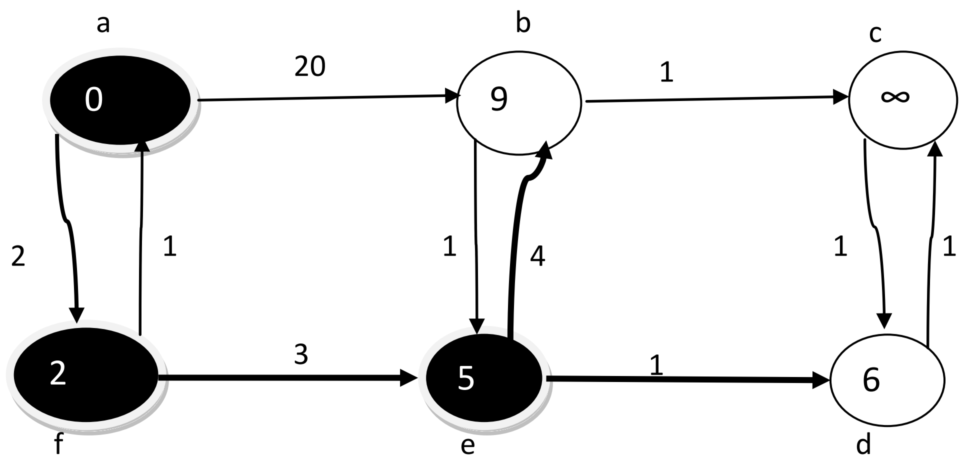


Fig.7: choose d ϵ V \ S,

while {b, c, d} ∉ S

choose d ϵ V \ S,

u = d because d.d = min {d.b, d.d} = min {9,6}

for all v = {c} ϵ vierus[d]

Relax (d,c,w)

c.d = 6+1 = 7, 7 < ∞

c.p = d

d.p = e

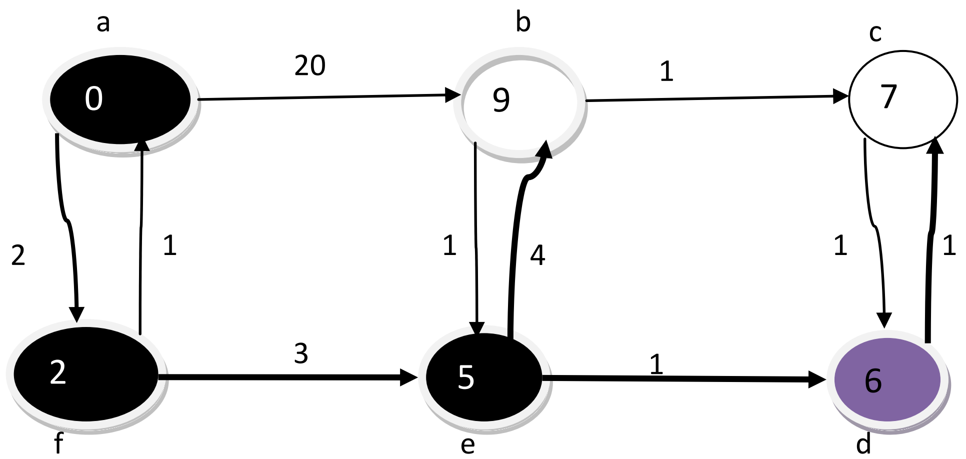


Fig.8: S = S ∪ d

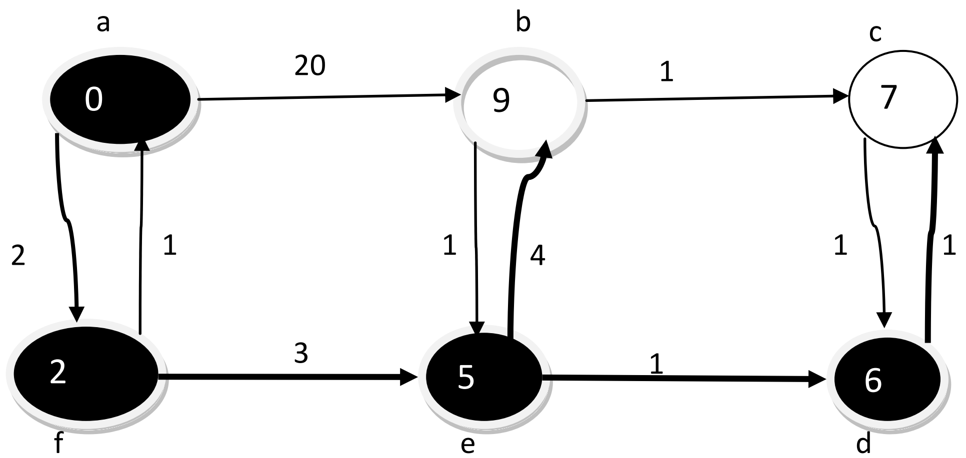


Fig.9: choose c ϵ V \ S

while {b,c} ∉ S

choose c ϵ V \ S,

u = c because d.c = min {d.c, d.b} = min {7,9}

for all v = ∅ ϵ vierus[c]

Relax (f,v,w)

c.p = d

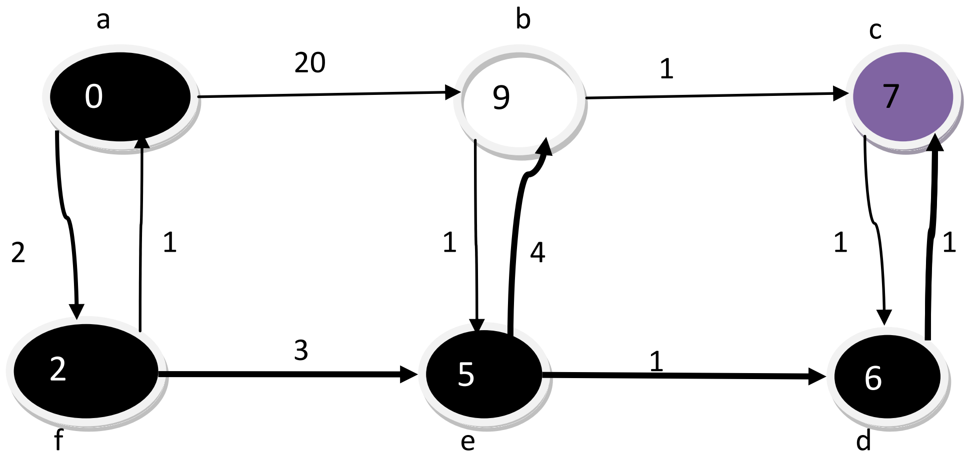


Fig.10: S = S ⋃ *c*

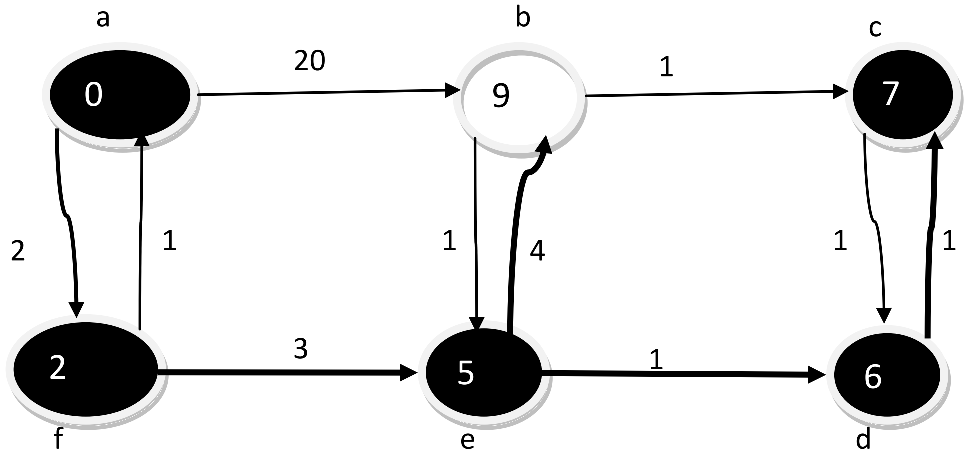


Fig.11: choose b ϵ V \ S

while {b} ∉ S

choose b ϵ V \ S,

u = b because d.b = min {d.b} = min {9}

for all v = ∅ ϵ vierus[c]

Relax (f,v,w)

b.p = e

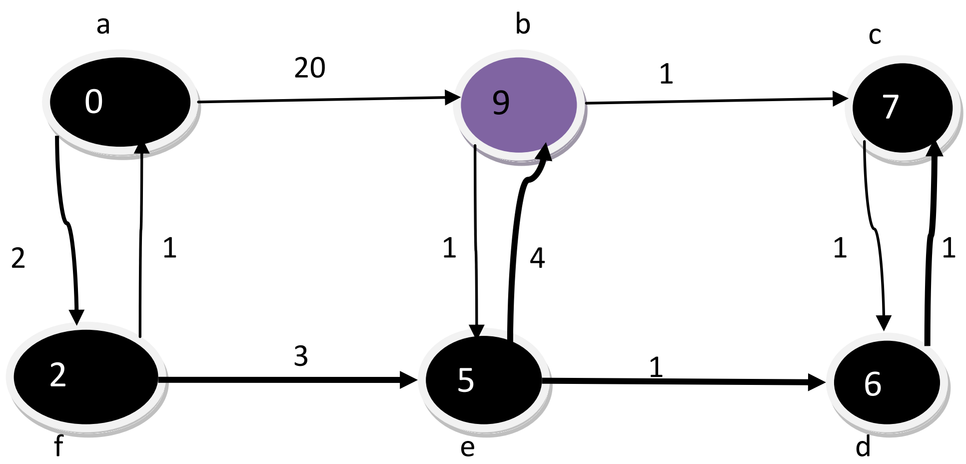
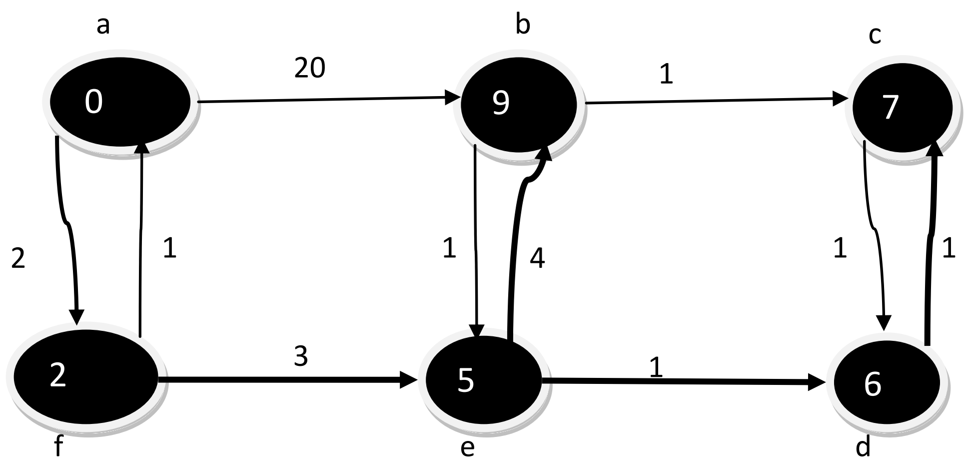


Fig.12: S = S ⋃ *b*



At the end S = {a, f, e, b, d, c}}

Shortest path:

```
a -- > f -- > e -- > b
              |
              V
              d -- > c
```