COURS_5 : LE PLUS COURT CHEMIN DANS UN GRAPHE 2

I. Algorithme de Dijkstra avec FILE :

Algorithme de Dijkstra :

```
dist[∞ for u in G]
P[-1 for u in G]
s =0
Q = []
Q.append(s)
dist[s] = 0

while Q :
        z = delete()
        for x in adj[z] :
                tense = d[z] + W[z][x]
                if dist[x] > tense :
                        dist[x] = tense
                        P[x] = z
                        Q.add(x)
```

Algorithme de Dijkstra :

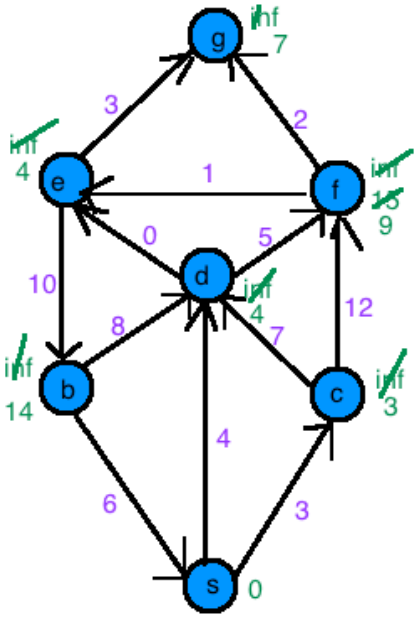
```
dist[∞ for u in G]
P[-1 for u in G]

Q = V
dist[s] = 0
P[s] = s

while Q :
        z = Q.deleteMin()
        for x in adj[z] :
                tense = d[z] + W[z][x]
                if dist[x] > tense :
                        dist[x] = tense
                        P[x] = z
```

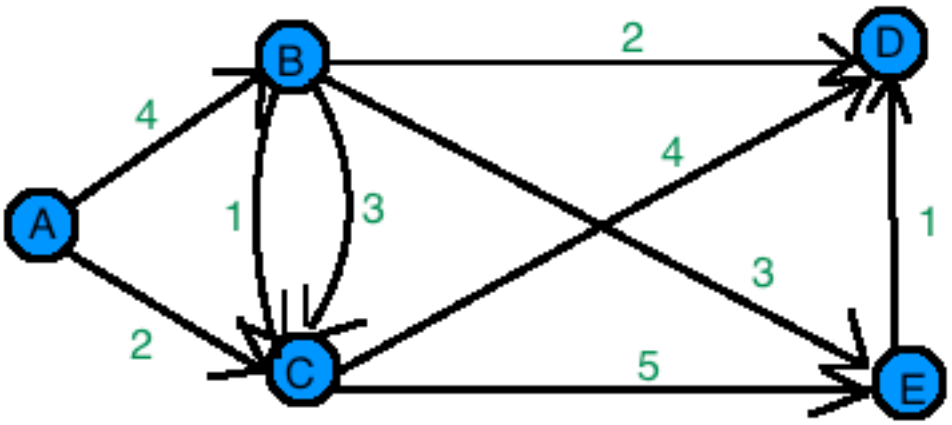
=> $o(V^2 + E)$

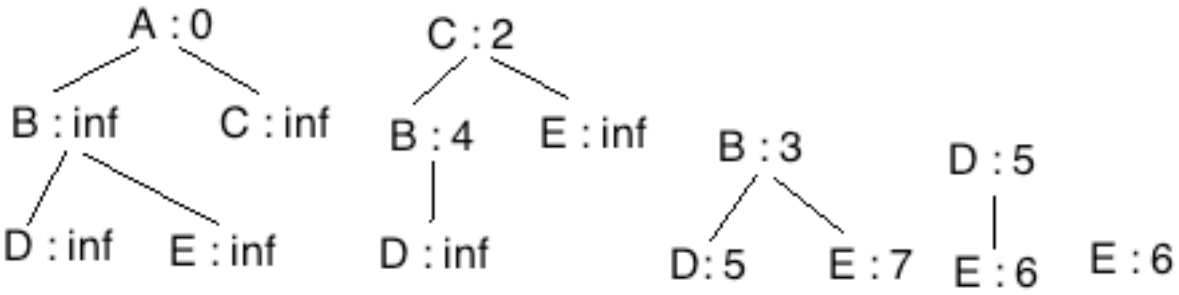
exemple :



Problème : a chaque fois on cherche le min pour le supprimer c'est pourquoi la complexité est aussi grande.

II. Algorithme de Dijkstra avec TAS :



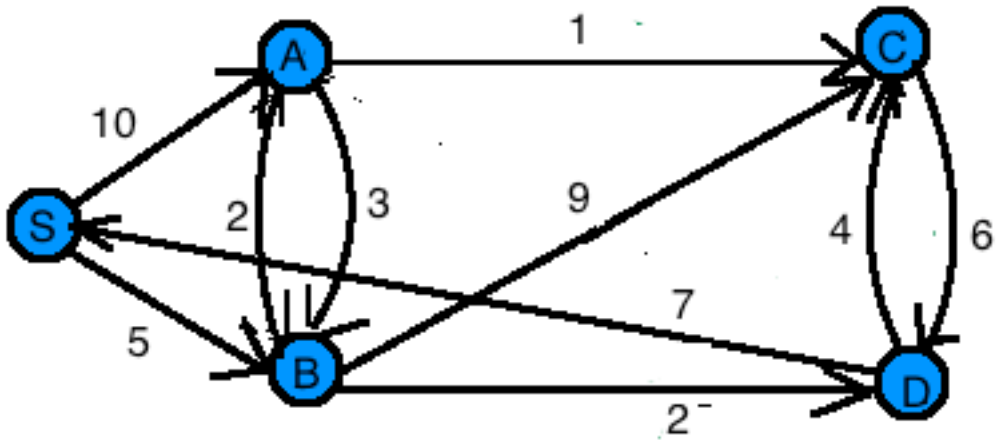


III. Module heapp sur python : (heap = tas en anglais)

heappop(Q)
heappush(Q,x)
heapify(Q)
heapreplace(Q)

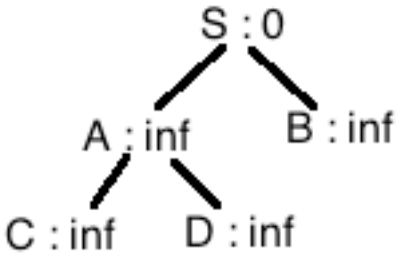
IV. Exercice :

faire le parcours du graphe suivant :

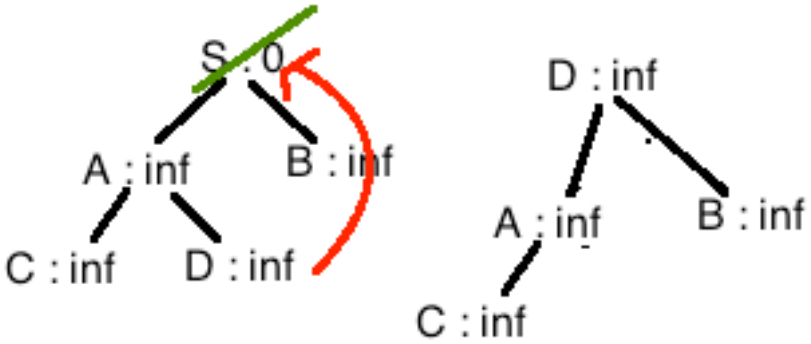


1. si Q = V
2. si Q = {s}

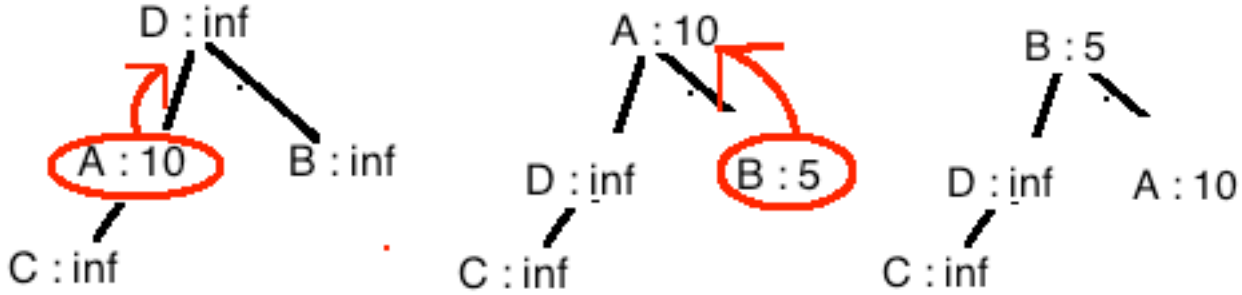
1.

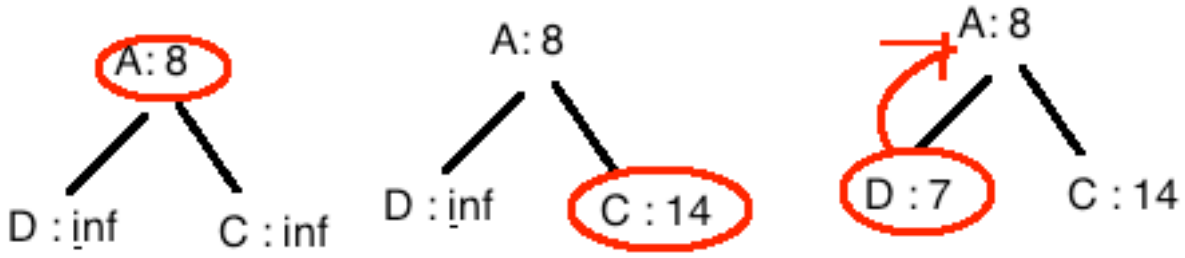


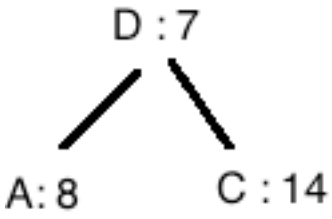
1. On supprime S



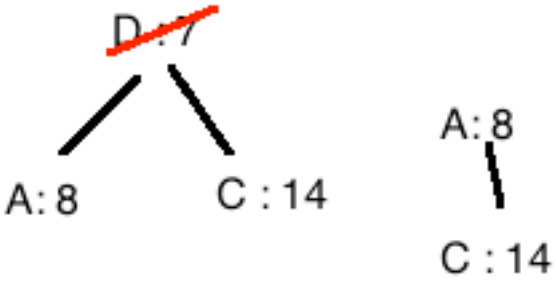
On modifie



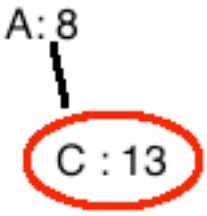




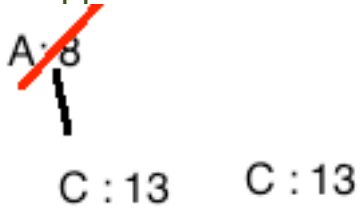
On supprime D



on modifie



On supprime A

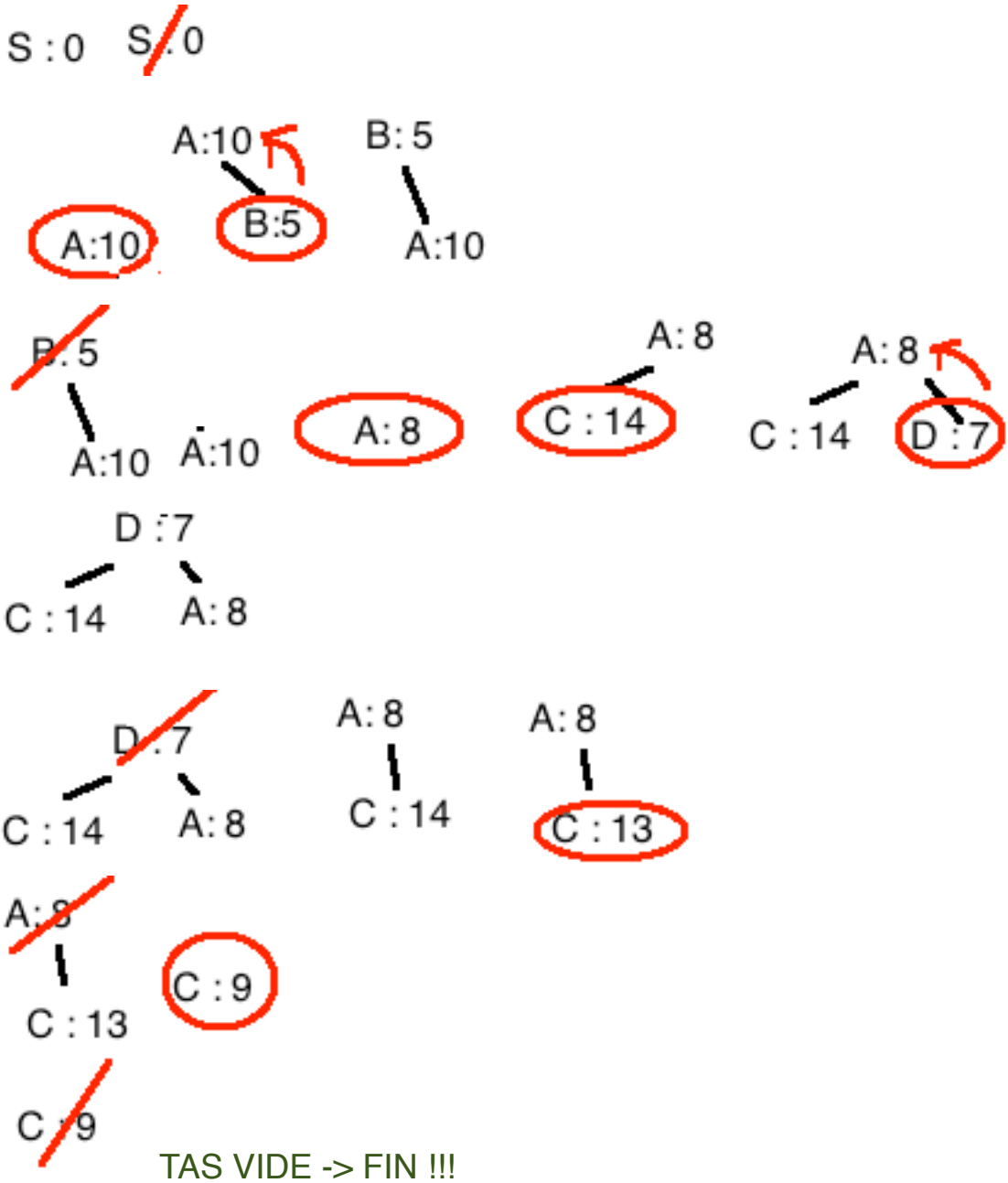


on modifie

on supprime C

TAS VIDE -> FIN !!!

2. SI Q = {S}



TAS VIDE -> FIN !!!