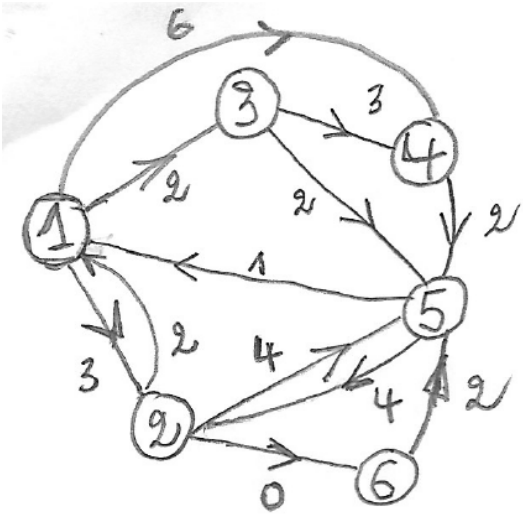
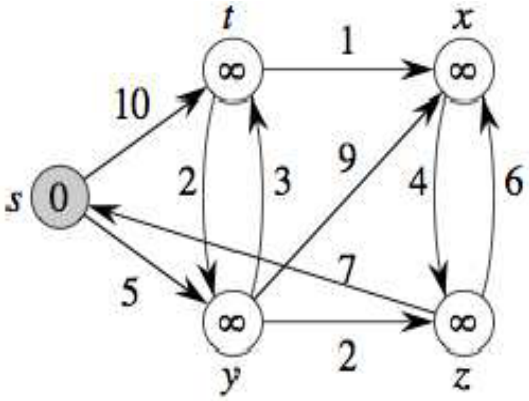
L3-INFO – INF6B
Structures discrètes
2020-2021

TD 02 - Algorithmes de plus courts chemins (DIJKSTRA et BELLMAN-FORD)

Exo #1. Algorithme de DIJKSTRA

Q1. Rappeler les conditions d'application de cet algorithme.





Q2. Appliquer l'algorithme de DIJKSTRA sur le graphe de gauche en prenant comme source (s=1). Pour cela, compléter :

- le tableau **D[t]** qui est le tableau des distances au sommet source s
- le tableau **PRED[t]** qui est le tableau des prédécesseurs

Visités	**D[1]**	**D[2]**	**D[3]**	**D[4]**	**D[5]**	**D[6]**
{}	0	inf	inf	inf	inf	inf
{1}	0	3	2	6	inf	inf

PRED	**1**	**2**	**3**	**4**	**5**	**6**
	nil	nil	nil	nil	nil	nil
	nil	1	1	1	nil	nil

Q3. Appliquer l'algorithme de DIJKSTRA sur le graphe de droite en prenant comme sommet source s. (exo. de révision à faire en dehors du TD ; ci-dessous la solution)

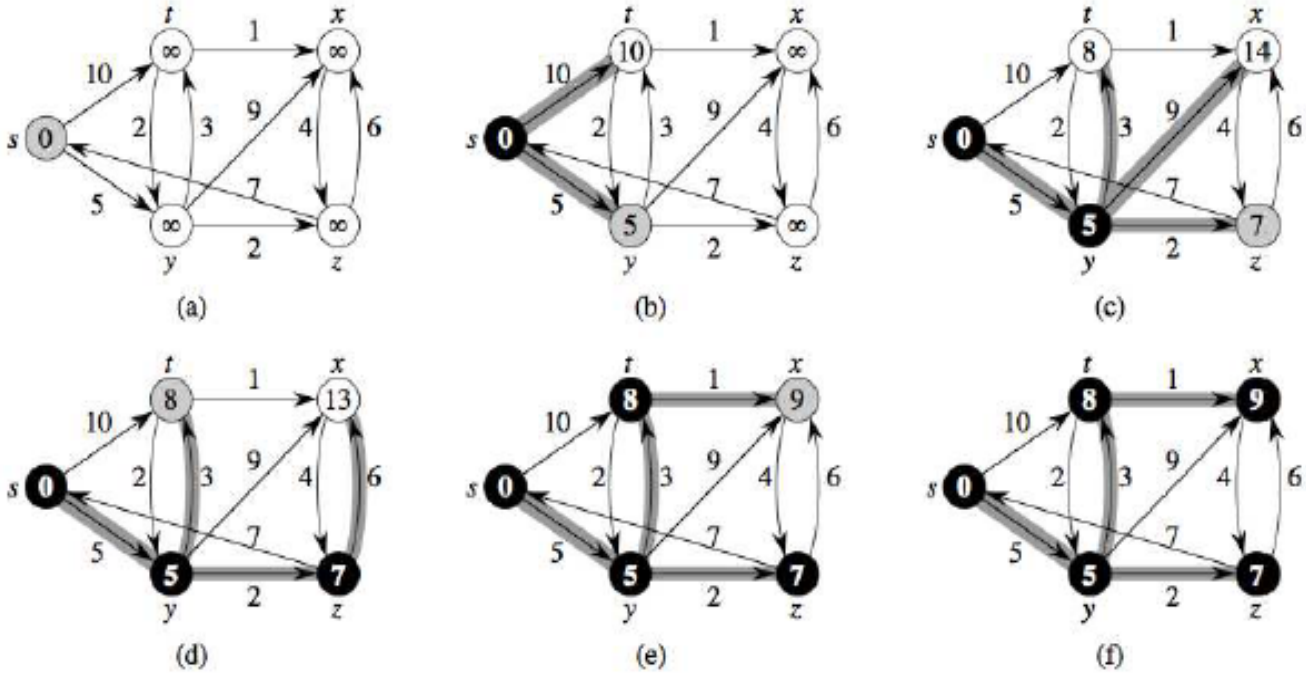
Evolution du tableau DistSource[u] au fur et à mesure des itérations

Iter	s	t	y	x	z
0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	10	5	$+\infty$	$+\infty$
2	0	8	5	14	7
3	0	8	5	13	7
4	0	8	5	9	7
5	0	8	5	9	7

Tableau des prédécesseurs

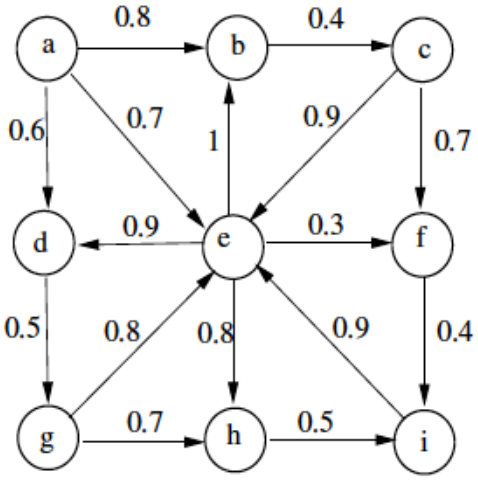
s	t	y	x	z
-1	y	s	t	y

Sélections successives du sommet à traiter et construction de l'arbre des plus courts chemins



Exo #2. Adapter l'algorithme de DIJKSTRA. On considère un réseau de télécommunication, composé d'émetteurs/récepteurs pouvant s'envoyer des messages, avec une certaine fiabilité de communication, i.e. une certaine probabilité pour que la communication ne soit pas interrompue.

On modélise ce problème à l'aide du graphe orienté valué suivant, où la valuation d'un arc est une valeur réelle comprise entre 0 et 1 indiquant la probabilité que la communication se passe sans souci.



Question : Comment déterminer le chemin le plus fiable pour envoyer un message d'un sommet source (ici le sommet a) vers un sommet cible (ici le sommet i) ?

On cherche un chemin pour lequel le produit des probabilités sera maximal. Pour cela, adapter l'algorithme de DIJKSTRA en apportant les corrections en bleu à l'algorithme ci-dessous :

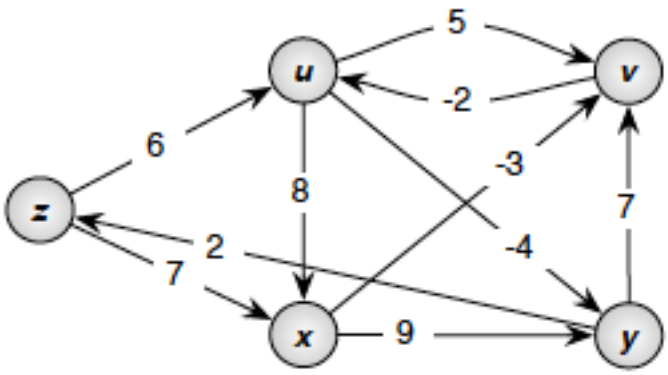
```
procédure DIJKSTRA_ADAPTE(G : graphe valué, s : sommet source) ;
début
DistSource[s] := ??? ;
pour tout sommet u ≠ s faire  DistSource[u] :=   ??? ;
pour tout sommet u faire  PRED[u] := nil ;
tantque  tous les sommets ne sont pas marqués  faire
        soit t0 un sommet non marqué tq   ???
        marquer le sommet t0 ;
        pour tout successeur non marqué t de t0  faire   Relacher(t0 -> t) ;
fttque
fin

procédure Relacher (arc u -> v) ;
début
        si  DistSource [v]   ???
            alors
                DistSource[v] :=  ???
                PRED[v] :=  u
        fsi
fin
```

Rq. Cet algorithme n'est correct que si les poids des arcs sont tous compris entre 0 et 1. Chaque fois que l'on ajoute un arc à un chemin, on diminue le coût total du chemin (ceci pour la même raison que L'algorithme de DIJKSTRA n'est correct que si tous les coûts sont positifs ou nuls).

Exo #3. Algorithme de BELLMAN-FORD

On considère le graphe orienté valué suivant :



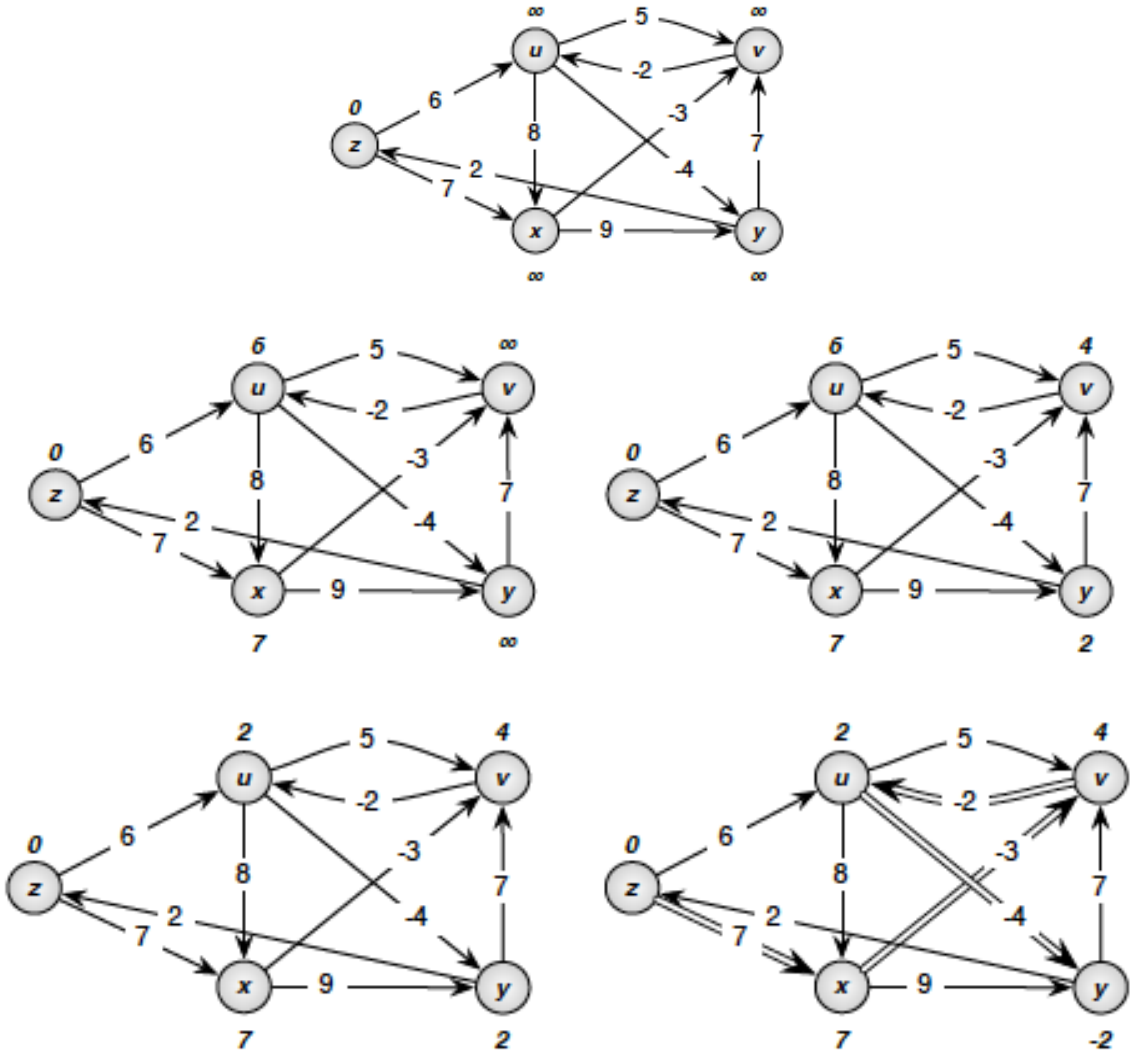
Q1. Appliquer l'algorithme de BELLMAN-FORD au graphe ci-dessus en prenant comme sommet source le sommet z.

Indications. *Durant l'application de l'algorithme, on considèrera les arcs dans l'ordre suivant : (u, v), (u, x), (u, y), (v, u), (x, v), (x, y), (y, v), (y, z), (z, u), (z, x)*

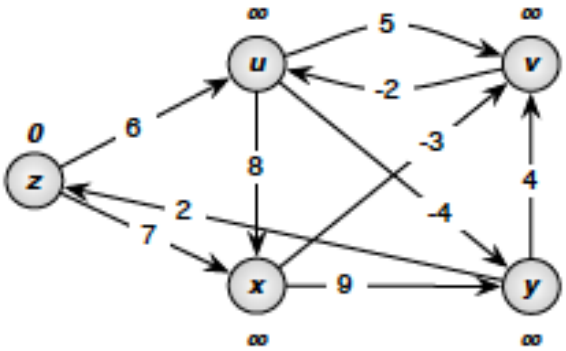
Q2. Même question mais en changeant le poids de l'arc (y, v) à 4.

Montrer qu'il existe un circuit absorbant ? Que peut-on en déduire

Eléments de correction pour la question Q1.



Eléments de correction pour la question Q2.



v-u-y-v est un circuit absorbant. Bellman-Ford détecte l'absence de PCC