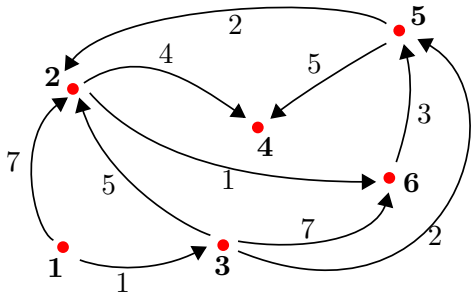
Recherche Opérationnelle et Optimisation
TP5 : Algorithmes de plus courts chemins

Responsable : Julien Ah-Pine

M1 Informatique 2017/2018

1 Exercice

Soit le graphe ci-dessous :



1. Calculer les plus courts chemins entre 2 et tous les autres sommets en développant à la main l'algorithme de Moore-Dijkstra.
2. Calculer les plus courts chemins entre 2 et tous les autres sommets en développant à la main l'algorithme de Ford-Bellman.

2 Quelques commandes utiles

1. Entrez et exécutez les commandes suivantes les unes après les autres :

```
pi=c(5,1,2,Inf,3,7)
i=which.min(pi)
i
pi[i]
```

   Que fait chacune de ces commandes ?

2. Entrez et exécutez les commandes suivantes les unes après les autres :

```
X=1:6
S=c(1,2)
Sb=setdiff(X,S)
pi[Sb]
j=which.min(pi[Sb])
Sb[j]
```

   Que fait chacune de ces commandes ?

3 Algorithme de Moore-Dijkstra

3. Ecrivez une fonction `Moore_Dijkstra` qui prend en entrée un ensemble de sommets `X`, une matrice d'adjacence pondérée (de poids non négatifs) `A`, un sommet `s` de `X` et qui donne en sortie, les longueurs des plus courts chemins entre `s` et tous les autres sommets de `X` obtenus par l'algorithme de Moore et Dijkstra vue en cours.

4. Testez votre implémentation sur l'exercice précédent puis sur les exemples du cours.

4 Algorithme de Ford-Bellman

5. Ecrivez une fonction `Ford_Bellman` qui prend en entrée un ensemble de sommets `X`, une matrice d'adjacence pondérée `A`, un sommet `s` de `X` et qui donne en sortie, les longueurs des plus courts chemins entre `s` et tous les autres sommets de `X` obtenus par l'algorithme de Ford et Bellman vu en cours.
6. Testez vos deux fonctions sur l'exemple du cours.
7. Testez votre implémentation sur l'exercice précédent puis sur les exemples du cours.

5 Utilisation des deux algorithmes

On sait que l'algorithme de Moore et Dijkstra est garanti de donner la solution optimale à condition que les poids des arcs sont positifs. On sait par ailleurs que l'algorithme de Ford et Bellman est plus général et peut tenir compte des poids négatifs. En revanche, le temps de traitement de l'algorithme de Ford et Bellman dépend de l'ordre de traitement des noeuds et a une complexité plus grande que l'algorithme de Moore et Dijkstra.

Si le graphe contient des arcs de poids négatifs on pourra utiliser l'algorithme de Moore et Dijkstra pour avoir une solution approchée dans un temps raisonnable. Pour avoir la solution optimale, on pourra alors donner en entrer de l'algorithme de Ford et Bellman l'ordre approché donné par l'algorithme de Moore et Dijkstra.

Appliquer cette procédure sur l'exemple suivant :