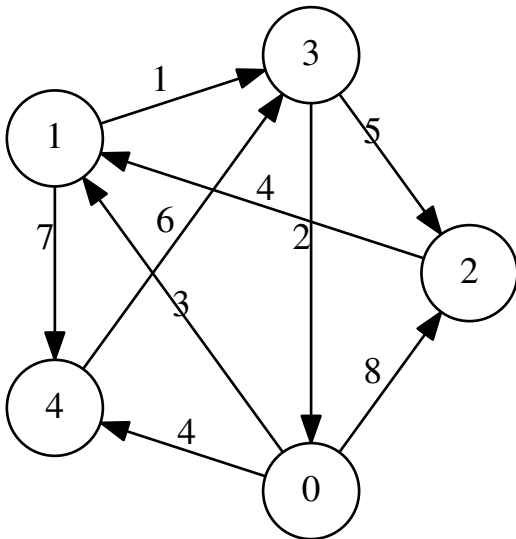


Plus courts chemins



Outils

- Rien à faire en Python. Réfléchir à comment on ferait en OCaml.
 - Pour représenter l'infini on pourra utiliser par exemple : `inf = float('inf')`
1. Définir le graphe ci-dessus sous forme de matrice d'adjacence (ici : liste de listes).
 2. Écrire une fonction permettant de créer une matrice vide/nulle de taille $n \times m$
 3. Écrire une fonction permettant de faire une copie de matrice

Pour les deux questions précédentes, attention aux histoires de pointeurs...

Floyd-Warshall

Pour l'algorithme de Floyd-Warshall, on travaille avec une **matrice d'adjacence**.

$$D_{i,j}^{(k+1)} = \min(D_{i,j}^{(k)}, D_{i,k}^{(k)} + D_{k,j}^{(k)})$$

4. Écrire l'algorithme de Floyd-Warshall (dans un premier temps, travaillez avec deux matrices si vous le souhaitez).

Dijkstra

Pour l'algorithme de Dijkstra, on travaille avec des **listes d'adjacences**

5. Écrire le graphe ci-dessus sous forme de listes d'adjacence (ici aussi : liste de listes).
6. File de priorité : ça existe en python ; voir <https://docs.python.org/3/library/heapq.html>. On ne peut pas mettre à jour, il faudra rajouter plusieurs fois le même noeud si besoin.

Dans la suite, on rappelle que la file de priorité contiendra des couples (**distance, noeud**).

7. Écrire l'algorithme de Dijkstra !

Bonus

8. Écrire des fonctions qui permettent de passer de matrice d'adjacence à liste d'adjacence, et inversement.
9. Programmez votre propre file de priorité avec un *tas – min*.
10. Floyd-Warshall et Automates : lire le sujet original (OCaml) pour découvrir cette technique intéressante !
11. On peut faire une file de priorité en OCaml en utilisant des foncteurs. . .