# Graphes - Plus cours chemin

## 1 Graphe Pondérés

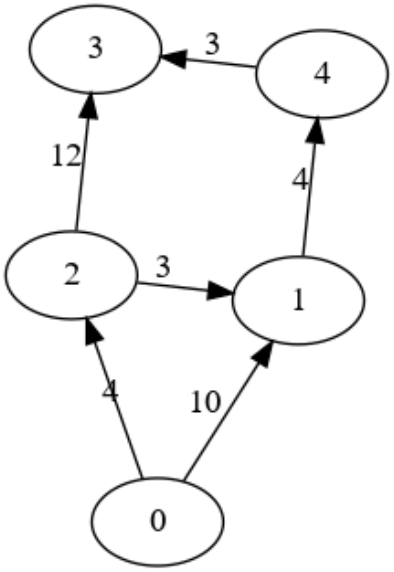


FIGURE 1 – `G1`

**Exercice 1.** *Liste d'adjacence*

Donnez la liste d'adjacence du graphe orienté pondéré $G_1$.

**Exercice 2.** *Matrice d'adjacence*

Donnez la matrice d'adjacence du graphe orienté pondéré $G_1$.

**Exercice 3.** *Voisin le plus proche*

Implémentez une fonction qui prend en entrée un graphe orienté pondéré représenté par sa liste d'adjacence et un sommet $u$ du graphe, et qui renvoie le voisins (successeur) $v$ de $u$ le plus "proche" (arête de poids minimal parmis les voisins) et le poids de l'arête en question.





## 2 Dijkstra

**Exercice 4.** *Sommet prioritaire*

Implémentez une fonction qui prend en entrée une liste de sommets et une liste de distances puis supprime (effet de bord) et renvoie le sommet ayant la plus petite distance dans la liste.

Les distances sont des entiers positifs et les sommets sont des entiers.

Pour supprimer un élément dans une liste la fonction `L.pop(i)` supprime le $i^{\text{ème}}$ élément de `L` et le renvoi. Sa complexité est en $O(n-i)$ avec $n$ la taille de la liste.

Sachant qu'il n'est pas important de conserver l'ordre des éléments de la liste `L` proposez une alternative en $O(1)$.

**Exercice 5.** *Relachement*

Implémentez une fonction prenant en entrée :
- Une liste de distances $D$. Les distances sont des entiers positifs avec eventuellement des poids infinis représentés par `None`.
- Une liste de parents $P$. Les parents sont des entiers positifs (sommets du graphe) avec eventuellement `None` pour les sommets qui n'ont pas de parent.
- Une arête pondérée sous la forme d'un triplé `(u,p,v)` représentant l'arête $u \xrightarrow{p} v$.

et relâchant par effet de bord l'arête $u \xrightarrow{p} v$.

**Exercice 6.** *Algorithme*

Implémentez l'algorithme de Dijkstra sur un graphe orienté pondéré

**Exercice 7.** *Reconstruction de chemin*

Implémentez une fonction qui prend une liste de parent telle que renvoyée par l'algorithme de Dijkstra et un sommets $v$, et qui renvoi le chemin de poids minimal de $s$, sommet de départ de l'algorithme de Dijkstra à $v$.

## 3 $A^*$

**Exercice 8.** *Sommet prioritaire avec heuristique*

Implémentez une fonction qui prend en entrée une heuristique $h$ (fonction Python), un sommet de destination $d$, une liste $L$ de couples de sommets et une liste de distances puis supprime (effet de bord) puis qui renvoie le sommet $u$ vérifiant $D[u] + h(s,u)$ est minimal.

**Exercice 9.** $A^*$

Implémentez l'algorithme de $A^*$ sur un graphe orienté pondéré pour une heuristique $h$ passée en paramètre.