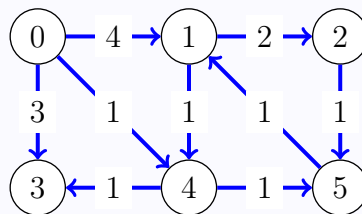

TP 27 - ALGORITHME DE DIJKSTRA & FILES DE PRIORITÉ

Exercice 1 (Déroulement de l'algorithme)

Appliquer l'algorithme de Dijkstra « à la main » pour déterminer les plus courts chemins, et leurs poids, depuis le sommet 0. Garder sur votre feuille des traces des différents états par lesquels passe la file de priorité aux différentes étapes.



Exercice 2 (Implantation d'une file de priorité)

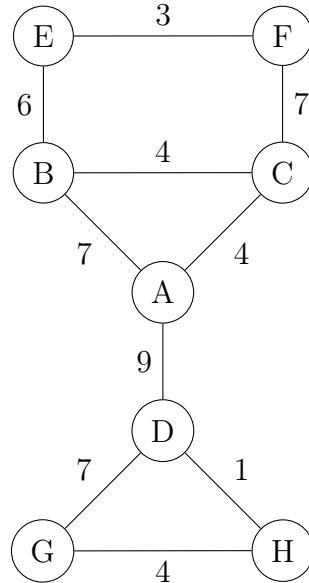
En utilisant des listes de paires (`element`, `priorité`) pour représenter les files de priorité ; codez les différentes opérations sur les files de priorité nécessaires pour coder l'algorithme de Dijkstra :

- Une fonction `file_prio()` qui ne prend pas d'arguments et qui renvoie une nouvelle file de priorité, vide.
- Une fonction `est_vide(f)` qui prend en entrée une file de priorité et renvoie en sortie un booléen qui indique si l'entrée est vide.
- Une fonction `enfiler_ou_maj(f, x, nouvelle_prio)` qui prend en entrée une file de priorité `f`, un élément `x` et une priorité `nouvelle_prio`. Cette fonction ne renvoie rien (dit autrement, elle renvoie `None`). Elle modifie en place la file de priorité, de sorte que
 - Si `x` n'est pas dans la file de priorité, il est rajouté, avec priorité `nouvelle_prio`.
 - Si `x` est dans la file, avec priorité `ancienne_prio`, alors il reste dans la file, avec comme priorité le minimum entre l'ancienne et la nouvelle priorité.
- Une fonction `defiler(f)` qui retire de la file, et renvoie, l'élément de `f` de plus faible priorité, ainsi que sa priorité. Le résultat renvoyé par cette fonction est donc une paire (`élément`, `priorité`).

Dictionnaires d'adjacence avec poids pour les arêtes

Dans ce TP, les graphes pondérés sont représentés par des dictionnaires qui à chaque sommet u associent une liste de paires de la forme (v, p) , où p est le poids de l'arête ou de l'arc reliant u à v .

Par exemple le graphe suivant :



Est représenté dans ce TP par la valeur python suivante :

```
{
    'A': [('B', 7), ('C', 4), ('D', 9)],
    'B': [('A', 7), ('C', 4), ('E', 6)],
    'C': [('A', 4), ('B', 4), ('F', 7)],
    'D': [('A', 9), ('G', 7), ('H', 1)],
    'E': [('B', 6), ('F', 3)],
    'F': [('C', 7), ('E', 3)],
    'G': [('D', 7), ('H', 4)],
    'H': [('D', 1), ('G', 4)]
}
```

Exercice 3 (Algorithme de Dijkstra)

Implantez l'algorithme de Dijkstra vu en cours.