

## Interrogation 2 : parcours de graphes et files

Durée : 20 minutes.

---

Nom :

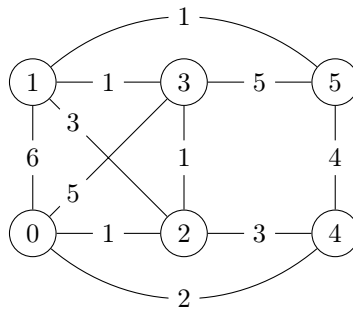
Prénom :

---

**Attention :** le sujet est recto-verso, n'oubliez pas de tourner la page.

### 1 Parcours de graphes

Exécutez l'algorithme de Dijkstra sur le graphe suivant à partir du nœud 0. Vous ne vous contenterez pas d'indiquer les plus courts chemins depuis le nœud 0, mais indiquerez bien à chaque itération de l'algorithme les informations stockées sur chaque nœud. Vous utiliserez de préférence la présentation en tableau vue en TD.



## 2 File d'attente

On propose l'implémentation d'une file d'attente via deux piles (in et out) suggérée en cours, une pour l'insertion et une pour le retrait.

**Procédure** insérer(file,  
élément)  
└ ajouter l'élément à file.in

**Fonction** retirer(file) → élément  
└ si file.out est vide **alors**  
└ └ **tant que** file.in n'est pas vide **faire**  
└ └ └ ajouter à file.out l'élément au sommet de file.in  
└ └ └ retirer le sommet de file.in  
└ retirer le sommet de file.out et le retourner

**fonctionnement :** pour chacune des étapes suivantes, indiquez le contenu des deux piles in et out. Précisez bien où se trouve le sommet de vos piles en les soulignant.

— insertion de 1 puis 2

— retrait

— insertion de 3 puis 4 puis 5

— retrait

— retrait

**complexité :** en supposant que les opérations sur les piles sont réalisées en temps constant, si  $n$  est le nombre d'éléments dans la file, quelle est la complexité du retrait de la file dans le meilleur et le pire des cas ?

**opérations par élément :** pour un élément inséré dans la file puis retiré par la suite, combien d'insertions et de retraits de pile concerneront cet élément ?

**coût d'une séquence :** étant donné une séquence d'actions contenant  $n$  insertions et  $n$  retraits, de telle sorte qu'aucun retrait de soit demandé sur une file vide, combien d'insertions et de retraits de pile seront réalisés au total ? Quel est donc la complexité moyenne d'un retrait dans ce cas ?