Université Claude Bernard – Lyon1 Licence Sciences et Technologies – L3
Algorithmique, Programmation et Complexité – LIFAP6

Interrogation 2 : parcours de graphes et analyse de complexité

Durée : 30 minutes.

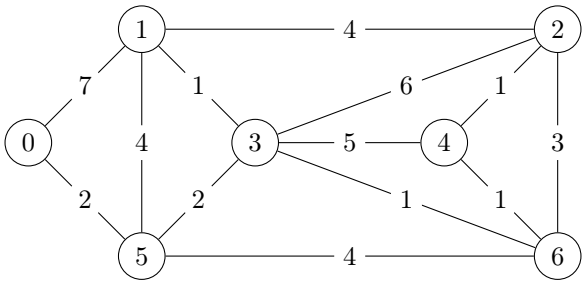
Nom :

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 ce qui se passe à chaque itération de l'algorithme, ainsi que les informations stockées sur chaque nœud.



2 Filimace

Cette partie vise à étudier une implémentation de file d'attente dans un tableau dynamique. Le principe consiste à utiliser deux indices pour marquer le début et la fin de la zone du tableau utilisée par la file, et à réallouer et déplacer les données de temps en temps pour éviter d'utiliser trop de place. L'insertion dans la file se fait en ajoutant un élément à la fin. Le retrait se fait en augmentant l'indice du début. Nous appellerons cette structure de données une Filimace. Une Filimace contient :

— tableau : le tableau contenant les valeurs mises dans la file, initialement alloué de capacité 1 ;
— capacite : le nombre de valeurs que peut stocker le tableau, initialisé à 1 ;
— debut : l'indice du premier élément de la file dans le tableau, initialisé à 0 ;
— fin : l'indice de la case du tableau juste après le dernier élément de la file, initialisé à 0.

Une Filimace est manipulée en utilisant les méthodes suivantes :

```
Fonction inserer(v) → rien
    si fin == capacite alors
        // la fin est au bout du tableau
        si debut > 0 alors
            // on recalle les éléments au début du tableau
            pour i de debut (inclus) à fin (exclu) faire
                tableau[i − debut] ← tableau[i]
        sinon
            // le tableau est plein, on l'augmente
            capacite ← capacite * 2
            allouer nouveau_tableau de taille capacite
            // copie de toute la file au début de nouveau_tableau
            pour i de debut (inclus) à fin (exclu) faire
                nouveau_tableau[i − debut] ← tableau[i]
            libérer tableau
            tableau ← nouveau_tableau
        fin ← fin − debut
        debut ← 0
    tableau[fin] ← v
    fin ← fin + 1
```

```
Fonction sommet() → valeur
    retourner tableau[debut]
```

```
Fonction retirer() → rien
    debut ← debut + 1
```

```
Fonction taille() → entier
    retourner fin − debut
```

Dans les questions qui suivent, les complexités vous sont demandées sous forme d'ordre de grandeur en utilisant les notations O, Θ, Ω. Vous considérerez que l'allocation d'un tableau, quelle que soit sa taille, est réalisée en temps $\Theta(1)$.

a) Quelle est dans le meilleur et le pire cas la complexité de la fonction inserer en fonction du nombre $n =$ fin − debut d'éléments dans la file ? Justifiez avec soin.

b) Soit une Filimace contenant déjà n éléments. Quelle est la pire séquence de n opérations supplémentaires (inserer, sommet, retirer, taille) que vous puissiez réaliser sur cette file en terme de complexité ? Vous êtes libre de choisir la capacité et le positionnement des données dans la Filimace initiale pour rendre cette séquence la pire possible. Justifiez avec soin.

c) Soit une séquence de n opérations de type inserer ou retirer réalisées sur une Filimace initialement vide. Quelle est dans le meilleur et le pire casla taille de la Filimace à l'issue de ces n opérations ?

Pour améliorer notre structure de données, au début de la fonction inserer, on remplace le test **si** debut > 0 par **si** debut $>$ capacite$/2$.

d) Soit une Filimace contenant n éléments et qui vient de subir insertion qui a provoqué un redimensionnement ou un décallage. Combien d'opérations au minimum faudra-t-il réaliser pour provoquer à nouveau un redimensionnement ou un décallage ?

e) Soit une séquence de n opérations réalisées sur une Filimace initialement vide, quelle est au pire le nombre de redimensionnements ou décallages qui seront causés par cette séquence ?