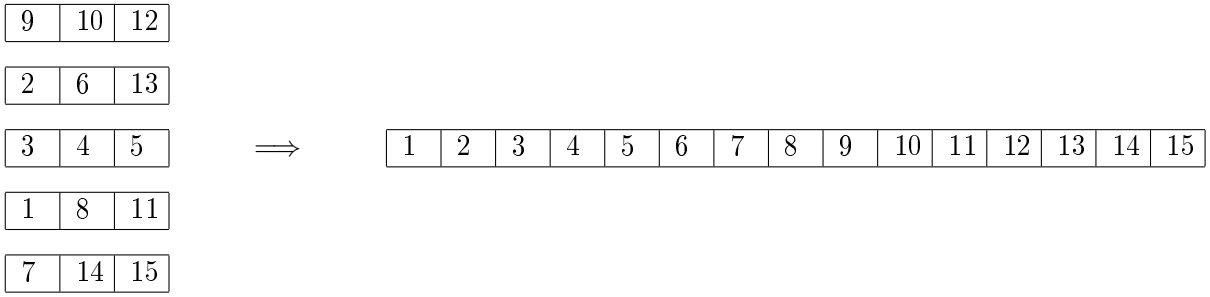
UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

LICENCE Informatique
Info0501 – 2020/2021

Travaux dirigés n° 4

Révisions et entraînement

Exercice 1 (Fusion de k tableaux triés)
Le problème de la fusion de k tableaux triés (*k-way merge problem*) consiste à fusionner k tableaux triés pour produire un unique tableau trié contenant les mêmes éléments. On note n le nombre total d'éléments qui est égal à la taille du tableau final et à la somme des tailles des k tableaux initiaux. Pour simplifier le problème, on suppose que tous les k tableaux sont de la même taille et ne sont pas vides, ce qui implique que $k < n$. La figure ci-dessous illustre un exemple où $n = 15$ et $k = 5$. À gauche, on retrouve les tableaux initiaux et à droite, le tableau final.



Il existe un algorithme efficace utilisant un tas min (une seule caractéristique différencie un tas min d'un tas max : dans la propriété de tas, le signe $\geq$ est remplacé par le signe $\leq$) pour résoudre ce problème dont les étapes sont les suivantes :

1. Créer un tableau de sortie de taille n
2. Créer un tas de taille k à partir des premières valeurs de chacun des k tableaux initiaux
3. Répéter les étapes suivantes n fois
   (a) Récupérer la valeur minimale du tas et la stocker dans le tableau de sortie
   (b) Remplacer la valeur minimale du tas par la prochaine valeur du tableau dans lequel la valeur minimale a été extraite (on suppose que chaque élément du tas possède un pointeur sur le tableau d'où provient sa valeur). Si ce tableau ne contient plus de valeur, remplacer la valeur minimale par ∞. Après avoir remplacé la valeur minimale, rétablir la propriété de tas du tas.

1°) En expliquant les éventuelles étapes intermédiaires et en donnant le résultat final, montrez/illustrez comment :

a) Construire efficacement le tas de l'étape 2.

b) Réaliser efficacement les 6 premières itérations de l'étape 3. Lors de la première itération, vous donnerez tout le détail du fonctionnement de l'algorithme.

2°) - Quel est le temps d'exécution de cet algorithme (en notation O) ? Justifiez.

- Si on n'utilise pas de tas min pour stocker les valeurs des k tableaux mais on travaille plutôt directement sur ces tableaux, on doit rechercher à chaque itération le minimum parmi tous les k tableaux et insérer ce minimum dans le tableau de sortie. Quel serait alors l'impact sur le temps d'exécution de l'algorithme ? Justifiez votre réponse en donnant le temps d'exécution de l'algorithme (en notation O) dans ce contexte et en le comparant avec celui trouvé en utilisant un tas min.

Exercice 2 (Transposé d'un graphe)

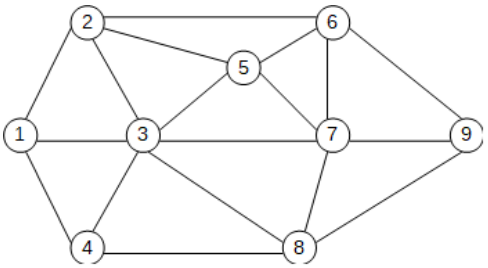
Le transposé d'un graphe orienté $G = (S, A)$ est le graphe $G^T = (S, A^T)$, où $A^T = \{(v, u \ \in SXS : (u, v) \in A\}$. Autrement dit, G^T est obtenu en inversant le sens de tous les arcs de G.

1°) Écrivez une fonction `transposé-liste` qui permet de calculer G^T à partir de G quand G est représenté par des listes d'adjacences. Analysez le temps d'exécution de votre algorithme.

2°) Écrivez une fonction `transposé-matrice` qui permet de calculer G^T à partir de G quand G est représenté par une matrice d'adjacences. Analysez le temps d'exécution de votre algorithme.

Exercice 3 (Parcours en largeur)

1°) Soit le graphe ci-contre, en expliquant votre démarche, donnez l'ordre obtenu en réalisant un parcours en largeur. Vous utiliserez le sommet 1 comme racine du parcours et supposerez que chaque liste d'adjacences est ordonnée par ordre croissant d'indice.

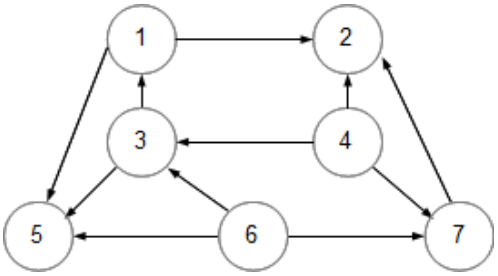


2°) Écrivez la procédure `parcours-largeur-matrice` qui effectue le parcours en largeur d'un graphe en utilisant une représentation par matrice d'adjacences.

Note : si votre algorithme nécessite des structures de données autres que la matrice d'adjacences du graphe, vous pouvez ne spécifier que la signature de leurs opérations sans les définir complètement.

Exercice 4 (Tri topologique)

Soit le graphe ci-contre, en expliquant votre démarche, donnez l'ordre obtenu en triant topologiquement les sommets. Vous considérerez les sommets en ordre croissant d'indice et supposerez que chaque liste d'adjacences est ordonnée par ordre croissant d'indice.



Exercice 5 (Arbres couvrants de poids minimal)

Soit le graphe ci-contre,

1°) En expliquant votre démarche et en spécifiant l'ordre de sélection des arêtes, donnez le résultat obtenu en appliquant l'algorithme de Kruskal.

2°) En expliquant votre démarche et en spécifiant l'ordre de sélection des arêtes, donnez le résultat obtenu en appliquant l'algorithme de Prim à partir du sommet 0.