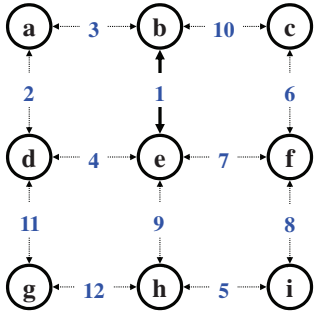
Feuille d'exercices no. 3

Structures Classe-Union

Exercice 1. Soit l'ensemble suivant de pièces (représentées par des cercles) reliées par des portes (représentées par des arcs numérotés indiquant leur ordre souhaité d'ouverture). Comme dans le cours (d'ailleurs c'est le même exemple), on veut ouvrir un nombre minimum de portes pour connecter toutes les pièces. Pour cela, on considère les portes une à une selon leur ordre d'ouverture, et on ouvre chaque porte si et seulement si elle connecte deux pièces qui ne sont pas encore connectées.



1. En supposant qu'au départ chaque pièce constitue une classe d'une structure Classe-Union (sans s'interesser à sa représentation), exécuter l'algorithme indiqué plus haut et lister dans leur ordre d'apparition les opérations $Classe()$ et $Union()$ qui sont effectuées. *Rappels.* (1) chaque classe est identifiée soit par un élément de la classe, soit par un numéro ; (2) une opération $Union(A, B)$ est effectuée sur deux classes ; pour réunir les classes de x et y, où x et y sont des pièces, il est nécessaire de savoir quelle est la classe à laquelle chaque pièce appartient, avant de réunir les deux classes.
2. Exécuter le même algorithme (on pourra juste suivre la liste des opérations à effectuer telle que listée ci-dessus) en supposant que la structure classe-union est représentée : a) par un tableau ; b) par un arbre pour chaque classe ; c) par un arbre pour chaque classe, en utilisant l'union par rang ; d) par un arbre pour chaque classe, en utilisant l'union par rang et la compression de chemins. Pour chaque cas, calculer la complexité de l'algorithme en notant par n le nombre de cases, et remarquant que le nombre m de portes ouvertes est en fait égal à $n - 1$ (la complexité ne dépendra donc que d'une seule variable).
3. Quelle est, d'après vous, la meilleure représentation en machine des arbres dans une structure Classe-Union de type arbre ? Pourquoi ?

Exercice 2. Dans cet exercice, nous considérons une structure de type Classe-Union avec union par rang et compression de chemins, contenant tous les éléments de l'ensemble $\{1, 2, \ldots, n\}$. Plus précisément, nous supposons que :

- la structure Classe-Union est représentée par le tableau $T[1..n]$ des parents de tous les éléments.
- le représentant de chaque classe est l'élément situé à la racine de l'arbre représentant la classe.
- l'appel à $Classe(i)$ retourne le représentant de la classe de i et effectue la compression de chemins le long du chemin entre i et la racine de sa classe. *Rappel.* Pour cela, l'élément i doit être accroché à la racine (et donc décroché de son père P). Son père P doit être accroché à son tour à la racine (avec tout son sous-arbre restant une fois que i a été décroché) et ainsi de suite jusqu'à la racine. Mais attention, la racine n'est pas connue à l'avance, donc les rattachements à la racine ne peuvent pas vraiment être faits pendant la montée dans l'arbre.
- l'union $Union(C_i, C_j)$ des classes contenant respectivement i et j est définie comme une union par rang, et est effectuée après l'appel à $Classe(i)$ et $Classe(j)$.

On demande (vous ajouterez, si besoin, des paramètres aux fonctions proposées) :

1. Soit $T = [10, 12, 12, 5, -, 5, 10, 5, 5, 12, 5, 5, 14, -, 6, 6]$.
   (a) Combien de classes a cette structure Classe-Union ?
   (b) Dessiner les arbres correspondant à chaque classe.
   (c) Est-ce qu'on peut obtenir la structure actuelle, avec des opérations $Union$ uniquement (les seuls appels à $Classe()$ étant ceux qui précèdent une $Union()$), à partir d'une structure de base où il y a $n = 16$ classes, chacune contenant exactement un élément parmi $1, 2, \ldots, 16$? Si oui, donner la suite d'opérations $Union$ qui doivent être effectuées. Si non, expliquer les raisons de votre réponse.
   (d) Indiquer le résultat de l'appel de $Classe(1)$ sur la structure T donnée, à la fois en mettant à jour le tableau T et en dessinant les arbres correspondant aux classes.
   (e) Indiquer le résultat de $Union(C_3, C_{13})$ (les classes de 3 et de 13) sur la structure T obtenue après l'appel du point précédent, à la fois en mettant à jour le tableau T et en dessinant les arbres correspondant aux classes.
2. Ecrire une fonction itérative, puis une fonction récursive, $Classe(i)$. A chaque fois, calculer combien de fois on traite chaque sommet du chemin qui lie i à la racine de sa classe.
3. Etant donnés deux indices i et j (pas forcément distincts) entre 1 et n, écrire la fonction $Union(C_i, C_j)$ qui fait l'union des classes de i et j.

Pour aller plus loin ...

Exercice 3. La compression des chemins est une technique efficace pour raccourcir les chemins, mais n'est pas la seule. Soit la technique suivante effectuée par $Classe(i)$: en remontant le chemin dans l'arbre depuis i vers la racine de la classe le contenant, chaque sommet z de ce chemin (y compris i lui-même) est attaché non pas à la racine, mais à son grand-père.

1. Quel effet est-ce que cela produit sur le chemin de i à sa racine ? Et sur les profondeurs des sommets qui étaient sur ce chemin ?
2. Ecrire l'algorithme pour $Classeprim(i)$ qui utilise cette technique.
3. Voyez-vous un avantage en nombre d'opérations effectuées par $Classeprim(i)$ par rapport à $Classe(i)$? Un inconvénient ?