

Université Paris-Est Créteil
IUT de Sénart-Fontainebleau – Département informatique
IS3 – Algorithmique Programmation
2012–2013

Partiel
Durée : 2 heures

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant (et celui-ci seulement) et les notes manuscrites.

Exercice 1.- (Date)

Écrire un applet Java qui demande une date sous la forme '19/12/12' dans un champ de texte et qui l'affiche sous la forme '19 décembre 2012'.

Exercice 2.- (Jeu de balles)

Écrire une application Java qui affiche un cadre (fermant) dans lequel apparaissent deux boutons « Avancer1 » et « Avancer 2 » ainsi que deux petits disques (de rayon 5 pixels), aux emplacements (0,0) et (0,150), appelées première et seconde balles.

Lorsqu'on appuie sur l'un des boutons, la balle correspondante avance des quantités dx et dy de valeur absolue 5.

Les balles se déplacent dans le rectangle de coin supérieur gauche (0,0), de largeur 200 et de hauteur 300. Lorsqu'une des balles rencontre une paroi, il y a rebond : l'une des quantités dx ou dy change de signe.

Lorsque les balles se rencontrent, il y a également rebond : les deux quantités $dx1$ et $dx2$ changent de signe.

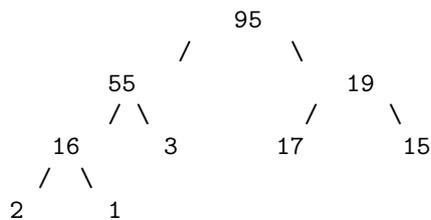
Exercice 3.- (Tri vertical)

Le tri vertical est une méthode efficace de tri, comparable au tri fusion, et donc nettement meilleure que le tri sélection ou le tri à bulle.

- 1^o) (Représentation séquentielle d'un arbre binaire complet)

Un arbre binaire est complet si tous ses niveaux, à l'exception possible du dernier, comportent le nombre maximum de nœuds et que ceux qui appartiennent au dernier niveau apparaissent aussi à gauche que possible.

Par exemple l'arbre (dont les nœuds sont étiquetés par des entiers) suivant est complet :



Un arbre binaire complet (étiqueté) se représente séquentiellement par un tableau : le type des éléments du tableau est celui des étiquettes ; sa dimension est le nombre de nœuds ; l'étiquette de la racine est l'élément d'index 0 ; les étiquettes des fils gauche et droit du nœud d'index i sont les éléments d'index $2i + 1$ et $2i + 2$.

Donner la représentation séquentielle de l'arbre complet ci-dessus.

- 2°) (**Arbre ordonné**)

a) Un arbre binaire complet étiqueté par les éléments d'un ensemble ordonné est ordonné si l'étiquette de chacun de ses nœuds est supérieure ou égale aux étiquettes de ses fils (lorsqu'ils existent).

L'arbre ci-dessus est-il ordonné ?

b) Insérer un élément dans un arbre ordonné consiste, dans une première étape, à ajouter un élément à la fin du tableau. L'arbre obtenu reste un arbre complet mais n'est pas en général ordonné.

Insérer 22 à l'arbre ci-dessus et représenter l'arbre obtenu. Cet arbre est-il ordonné ?

c) Insérer un élément consiste donc, dans une seconde étape, à réordonner de façon à obtenir un arbre ordonné : on compare l'élément ajouté à son père et échange les deux valeurs si le fils est plus grand que le père puis on continue de même. Le processus se termine lorsqu'on a pas besoin d'échanger ou lorsqu'on arrive à la racine.

Réordonner l'arbre obtenu ci-dessus en ajoutant 22. On présentera à la fois le tableau et la représentation sous forme d'arbre.

d) La méthode de tri vertical consiste à partir de l'arbre vide, d'insérer les éléments du tableau à trier un par un à cet arbre. On obtient un arbre ordonné dont l'élément le plus grand du tableau se retrouve à la racine de l'arbre (donc l'élément d'index 0 de sa représentation séquentielle). « Retirer la racine » consiste à échanger la racine et le dernier élément du tableau et à considérer le tableau dont la dimension effective est décrémentée d'une unité. En retirant la racine, on obtient un arbre complet mais qui n'est pas ordonné en général.

Représenter l'arbre obtenu en retirant la racine de l'arbre ci-dessus. Cet arbre est-il ordonné ?

e) « Retirer la racine » consiste donc, dans une seconde étape, à réordonner l'arbre obtenu de façon à obtenir un arbre ordonné : on compare la (nouvelle) racine à ses deux fils (tout au moins ceux qui existent) ; si elle est inférieure à l'un de ses fils, on l'échange avec le fils qui a la plus grande valeur puis on continue de même. Le processus se termine lorsqu'on a pas besoin d'échanger ou lorsqu'on arrive à une feuille.

Réordonner l'arbre obtenu en d). On présentera à la fois le tableau et la représentation sous forme d'arbre.

f) Le tri vertical consiste évidemment à retirer les racines successives jusqu'à obtenir l'arbre vide.

Trier le tableau originel (obtenu au 1°) en utilisant le tri vertical.

- 3^o) (**Codage du tri vertical en Java**)

Écrire une classe Java **Tree** dont les attributs sont un tableau d'entiers et la dimension effective de celui-ci. Ses méthodes sont :

- un constructeur permettant de spécifier la dimension maximum du tableau ;
- une méthode `insert()` permettant d'insérer un entier (dans l'arbre complet ordonné) ;
- une méthode `tri()` permettant de trier le tableau (en utilisant la méthode du tri vertical) ;
- une méthode `affiche()` des éléments du tableau séparés par deux espaces.

- 4^o) Écrire un programme de test insérant, un à un, les entiers 2, 55, 15, 95, 3, 23, 17, 1, 16 et 5, triant le tableau et affichant le tableau trié.

Documentation

1 Sous-mot

La méthode :

```
String substring(int, int)
```

de la classe **String** permet de récupérer la sous-chaîne comprise entre les indices déterminés par le premier argument et le second argument moins un. Les indices commencent à 0.

2 Les plaques ovales

La classe **Graphics** possède la méthode :

```
fillOval(int x, int y, int width, int height);
```

permettant de dessiner une plaque ovale, où `x` et `y` sont les coordonnées du coin en haut à gauche du rectangle inscrivant l'ovale, `width` et `height` la largeur et la hauteur de ce rectangle.