Université Paris XII
UFR de Sciences et Technologie
Département Informatique
Master SSI deuxième année
2006/2007

EXAMEN MELO

2 heures

Exercice 1.- (Sans document, 1 point)

Expliquer pourquoi les ASM sont, d'un de vue théorique, très bien adaptées à la modélisation. On pourra comparer, par exemple, avec les machines de Turing ou d'autres modèles.

Exercice 2.- (**Remplissage d'une région**, documents permis)

Un problème courant en graphisme est de **remplir une région** *: on a une image, on désigne un pixel de celle-ci et on veut que la couleur de la région à laquelle appartient ce point change. Dans le cas d'image monochrome, on changeait la couleur de celle-ci ; dans le cas d'image couleur, on indique une nouvelle couleur et toute la région doit prendre cette nouvelle couleur.*

À un moment ou un autre une **image** *sera un tableau bidimensionnel d'entiers : au pixel de coordonnées x et y de couleur c, on fera correspondre :*

$$tab[x, y] = c.$$

Qu'est-ce qu'une région ? Les spécialistes en sont venus à distinguer les 4-régions, les 8-régions ou d'autres notions. Dans le cas le plus simple des 4-régions, chaque pixel, de coordonnées (x, y), possède quatre voisins immédiats, ceux de coordonnées $(x, y-1)$ (au-dessus), $(x-1, y)$ (à gauche), $(x+1, y)$ (à droite) et $(x, y+1)$ (en-dessous), dans le cas bien entendu où ces pixels appartiennent à l'image.

Considérons l'image suivante :

```
2 0 4 6 0
0 0 1 4 0
5 0 0 8 0
1 0 0 8 0
0 1 0 0 0
```

Remplir la région à laquelle appartient le pixel de coordonnées $(1, 1)$, d'ancienne couleur 0, avec la couleur 7 nous conduit à l'image suivante :

```
2 7 4 6 7
7 7 1 4 7
5 7 7 8 7
1 7 7 8 7
0 1 7 7 7
```

Remarquons que le pixel en bas à gauche n'a pas changé de couleur (il aurait changé de couleur avec les 8-régions).

L'algorithme pour résoudre ce problème utilise une pile pour garder trace des pixels qu'il faut exaaminer :

```
Empiler les coordonnées du point de départ.
Tant que la pile est non vide :
    dépiler un point P
    si la couleur de P est l'ancienne couleur
       empiler les (au plus) quatre voisins de P
       faire prendre à P la nouvelle couleur
```

- 1^o) (7 points) Implémenter en AsmL :

1. (0.5 point) Une classe `Point` avec deux attributs entiers (les coordonnées x et y).

2. (0.5 point) Une classe `PileElt` dont les instances sont les éléments d'une pile de points.

3. (1.5 points) Une classe `Pile` de points dont, classiquement, l'attribut est le dessus de la pile (de type `PileElt`), et dont les méthodes sont `isEmpty()`, `push()` et `pop()`.

   [*Il peut être plus intéressant que la méthode* `push()` *aient deux arguments entiers plutôt qu'un argument de type point.*]

4. Une classe `Image` dont les attributs sont un entier `xmax` (le nombre maximun effectif de pixels horizontaux), un entier `xmax` (le nombre maximum effectif de pixels verticaux) et un écran `screen` de type suite de suites d'entiers.

   [*L'image :*

   ```
   4 6
   0 0
   ```

   est représentée par (2, 2, [[4, 6], [0, 0]]).]

   Les méthodes sont `lireImage()`, `afficheImage()` et `fill()`.

   (1 point) La méthode `lireImage()` sera très rudimentaire : on demande à l'utilisateur les valeurs de `xmax`, `ymax` et des couleurs de chaque pixel un par un.

   (1 point) L'affichage ressemblera aux exemples ci-dessus : on affiche les couleurs de chacun des pixels de la première ligne (séparées de deux espaces), puis de la deuxième ligne et ainsi de suite.

(1 point) La méthode `fill()` prend quatre arguments entiers : les coordonnées du point déterminant la région, l'ancienne couleur (de façon redondante) et la nouvelle couleur. Elle implémente l'algorithme décrit ci-dessus.

5. (0.5 point) Un programme de test.

- 2^o) (1 point) Doit-on, dans la méthode `fill()`, traiter les quatre voisins séquentiellement ou en parallèle ? Justifier votre réponse.

- 3^o) (2 points) Réécrire la méthode `afficherImage()` sous forme normale.