TP5 : Tris sur des tableaux et des listes

Christophe Rose

Vendredi 28 mars 2008

christophe.rose@ens.fr

http ://www.eleves.ens.fr/home/rose/caml

Dans ce TP, nous allons implémenter certains algorithmes de tri sur des tableaux et des listes.

1 Algorithmes de tri

Question 1. Écrivez une fonction `tab_test` de type `int -> int vect` qui crée un tableau de taille donnée rempli d'entiers aléatoires. On utilisera la commande `random__int 1000000000;;`. Écrivez de même la fonction `liste_test`.

Question 2. Écrivez une fonction `tab_trie` de type `'a vect -> bool` qui détermine si un tableau est trié dans l'ordre croissant. Écrivez de même la fonction `liste_triee`.

Question 3. Écrivez une fonction `algo_tab_ok` de type `('a vect -> 'a vect) -> int -> bool` qui vérifie si une fonction sensée trier un tableau fonctionne sur un tableau aléatoire de taille donnée. Écrivez de même la fonction `algo_liste_ok`.

La fonction suivante sera utilisée dans tous les algorithmes de tri sur les tableaux.

Question 4. Écrivez une fonction `echange` telle que `echange v a b` échange les valeurs `v.(a)` et `v.(b)`.

2 Tri insertion

Pour effectuer le tri insertion, on considère successivement tous les éléments d'un tableau. On les insère un par un dans un autre tableau, de sorte que ce dernier tableau soit toujours trié. À la fin, le deuxième tableau est trié et contient les mêmes éléments que le premier.

3	*1*	*5*	*7*	*4*	*8*
3	*1*	*5*	*7*	*4*	*8*
1	**3**	*5*	*7*	*4*	*8*
1	**3**	**5**	*7*	*4*	*8*
1	**3**	**5**	**7**	*4*	*8*
1	**3**	**4**	**5**	**7**	*8*
1	**3**	**4**	**5**	**7**	**8**

On commence par utiliser des listes.

Question 5. Programmez une fonction `insere` de type `'a -> 'a list  -> 'a list` qui insère un élément à la bonne position dans une liste suposée triée.

Question 6. Programmez une fonction `tri_insert_liste` qui prend en argument une liste et qui renvoie la liste triée dans l'ordre croissant.

Question 7. Vérifiez que la fonction ci-dessus s'exécute convenablement sur certaines listes, de tailles variant entre 0 et 4000000.

Question 8. Programmez la fonction `tri_insert_tab` et vérifiez qu'elle s'exécute convenablement.

3 Tri sélection

Pour effectuer le tri sélection, on cherche le plus petit élément du tableau, qu'on déplace en première position. Puis on cherche le deuxième plus petit, qu'on déplace en deuxième position. À la fin, tous les éléments sont à leur place.

3	*1*	*5*	*7*	*4*	*8*
1	*3*	*5*	*7*	*4*	*8*
1	**3**	*5*	*7*	*4*	*8*
1	**3**	**4**	*7*	*5*	*8*
1	**3**	**4**	**5**	*7*	*8*
1	**3**	**4**	**5**	**7**	*8*
1	**3**	**4**	**5**	**7**	**8**

Question 9. Programmez une fonction `plus_petit` telle que `plus_petit v k` trouve l'indice du plus petit élément de `v` d'indice supérieur ou égal à k.

Question 10. En déduire une fonction `tri_select_tab` de type `'a vect -> 'a vect` qui effectue le tri sélection.

Question 11. Vérifiez que la fonction ci-dessus s'exécute convenablement sur certains tableaux, de tailles variant entre 0 et 4000000.

Question 12. Programmez la fonction `tri_select_liste` et vérifiez qu'elle s'exécute convenablement.

4 Tri à bulles

Pour effectuer le tri à bulles, on considère tous les couples d'éléments successifs, en partant du début du tableau. Si un couple n'est pas trié, on les échange. Lorsqu'on a parcouru tout le tableau, on sait que le plus grand élément est à la fin. À la fin, tous les éléments ont remonté à leur position.

3	*1*	*5*	*7*	*4*	*8*
1	*3*	*5*	*7*	*4*	*8*
1	*3*	*5*	*4*	*7*	*8*
1	*3*	*5*	*4*	**7**	**8**
1	*3*	*4*	*5*	**7**	**8**
1	*3*	*4*	**5**	**7**	**8**
1	**3**	**4**	**5**	**7**	**8**

Question 13. Programmez la fonction `tri_bulle_tab`, et vérifiez qu'elle s'exécute convenablement.

Pour effectuer le tri à bulles sur des listes, il faut faire passer la bulle de droite à gauche. Ce sont donc les éléments à gauche qui sont triés en premier.

Question 14. Programmez une fonction `passage_bulle` qui fait le passage d'une bulle sur une liste. En déduire la fonction `tri_bulle_liste`.

5 Matrices

Une matrice de type `'a` est un tableau de tableaux de type `'a`.

On ne peut pas créer une matrice simplement par les commandes suivantes :

```
# let A =
  let v = make_vect nb_colonnes 0 in
  make_vect nb_lignes v ;;
```

Sinon, la commande `A.(0).(0)<-1` serait l'équivalent de `v.(0)<-1` et modifierait toute la première ligne et pas seulement le premier élément de la matrice.

Il faut donc recréer avec une boucle tous les vecteurs ligne.

```
# for i = 0 to (nb_lignes - 1) do
  let v = make_vect nb_colonnes 0 in
  A.(i)<-v;
  done;
  ;;
```

Question 15. Écrivez une fonction `dim` de type `'a vect vect -> int * int`, qui prend un matrice en argument et qui renvoie son nombre de lignes et son nombre de colonnes. Si l'argument n'est pas une matrice (pensez au triangle de Pascal), la fonction renvoie `(-1, -1)`.

Question 16. Écrivez des fonctions pour les opérations suivantes sur les matrices : addition, multiplication, transposée, trace, déterminant, inverse. On utilisera le type `float vect vect`.

6 Graphiques

Pour utiliser les fonctions graphiques de Caml, il faut charger le module correspondant avec la commande `#open "graphics";;` puis créer une fenêtre avec `open_graph "640x480";;`.

Pour changer la couleur utilisée, on utilise la fonction `set_color` suivi de la couleur (`black`, `white`, etc.). La fonction `plot x y` dessine un point en (x, y).

Question 17. Avec les fonctions `int_to_float` et `float_to_int`, dessinez une courbe sinusoïdale.

On part d'une matrice 640x480 remplie de booléens, en utilisant la convention `true=black` et `false=white`. Un agent (appelé *fourmi de Langton*) se déplace dans la matrice avec une direction donnée (nord/ouest/sud/est).

Lorsqu'il tombe sur une case noire, il la repeint en blanc et tourne à gauche. Lorsqu'il tombe sur une case blanche, il la repeint en noir et tourne à droite. Après avoir tourné, il se déplace d'une case en avant.

Question 18. Programmez une fonction `fourmi` qui dessine le trajet d'un tel agent, en partant du centre, et sur une matrice entièrement blanche. La fonction s'arrête lorsque la fourmi a atteint l'un des bords.

Question 19. Modifiez la fonction ci-dessus pour que l'agent commence sur une matrice donnée en argument, et puisse se déplacer comme dans un tore.