Travaux Dirigés d'algorithmique

Cours d'analyse, algorithmique

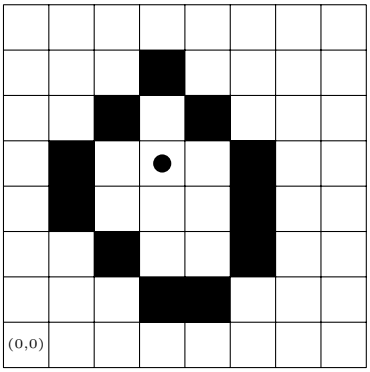
—Master 2 CCI—

Algorithmes, Révisions

Thèmes abordés : récursivité, algorithmes de tris et complexité, recherche d'éléments dans un tableaux.

▸ **Exercice 1. (Fonction récursive)**

On considère un tableau à deux dimensions contenant des cases noires ou blanches. Le tableau est initialement le suivant (on considère que $(0, 0)$ est en bas à gauche).



On exécute l'algorithme `noircir` suivant :

```
Algorithme noircir(x, y)

si tab[x,y] = blanc alors
   tab[x,y] <- noir
   noircir(x, y-1)
   noircir(x, y+1)
   noircir(x-1, y1)
   noircir(x+1, y)
```

```
Algorithme noircir2(x, y)

si tab[x,y] = blanc alors
   noircir2(x, y-1)
   noircir2(x, y+1)
   noircir2(x-1, y1)
   noircir2(x+1, y)
   tab[x,y] <- noir
```

1. On appelle l'algorithme `noircir` avec les coordonnées de la case contenant un point. Quel est l'état du dessin à la fin ?
2. Que pensez vous de l'algorithme `noircir2` ?

Les questions suivantes concernent l'algorithme `noircir`.

3. Dire dans quel ordre les cases ont-elles été noircies.
4. Montrer que le nombre d'appels récursifs est $1 + 4n$ où n est le nombre de cases noircies.
5. En déduire la complexité de cet algorithme.

▸ **Exercice 2. (Tri par sélection/échange)**

Soit T un tableau de taille n. Pour trier T, on peut procéder de la manière suivante :
- rechercher le plus petit élément de T, puis, s'il n'est pas en position 0, l'échanger avec l'élément d'indice 0 ;
- rechercher le second plus petit élément de T (c'est-à-dire le plus petit parmi les éléments d'indice plus grand que 1), puis, s'il n'est pas en position 1, l'échanger avec l'élément d'indice 1 ;
- rechercher le troisième plus petit élément de T (c'est-à-dire le plus petit parmi les éléments d'indice plus grand que 2), puis, s'il n'est pas en position 2, l'échanger avec l'élément d'indice 2 ;
- continuer de cette façon jusqu'à ce que le tableau soit entièrement trié.

Cette méthode est nommée tri par sélection/échange.

1. Écrire un algorithme pour chercher la position d'un plus petit élément d'un tableau (non trié).
2. En déduire un algorithme pour le tri par sélection/échange.
3. Quel est le nombre de comparaisons d'éléments du tableau effectuées par cet algorithme ? Quelle est sa complexité ?
4. Quel est dans le cas le pire, le nombre de copies d'éléments du tableau effectuées par cet algorithme ?
5. Expliquer pourquoi cet algorithme est intéressant pour trier des éléments dont la comparaison est rapide mais la copie lente.

▸ **Exercice 3. (Calcul des deux maximums d'un tableau).**

1. Écrire un algorithme qui retourne le plus grand élément et le deuxième plus grand élément d'un tableau ainsi que leur position.
2. Écrire un algorithme qui étant donné un tableau de `double` et deux `double` `min` et `max` répond si tous les entiers du tableaux sont dans l'intervalle `[min,max]`.