Initiation à l'algorithmique

TD 5

Dans les exercices qui suivent on manipule des tableaux de taille NMAX et on veut les trier dans l'ordre croissant.

Exercice 5.1

1. Faire tourner les trois méthodes élémentaires de tri étudiées en cours (tri sélection, tri insertion et tri à bulle), sur un tableau contenant la suite : $5, 9, 1, 5, 2, 8, 3$.
   Pour chacun des trois algorithmes, compter le nombre de comparaisons et le nombre d'échanges d'éléments de la suite.
2. L'un des trois algorithmes de tri peut être amélioré de façon à arrêter son exécution quand la suite à trier est ordonnée. Lequel ? Le modifier dans ce sens.

Exercice 5.2

Application des algorithmes de tri à des suites particulières :

1. Laquelle des trois méthodes de tri (tri sélection, tri insertion et tri à bulle), appliquée à un tableau contenant une suite croissante d'entiers (donc déjà ordonné), se termine le plus rapidement ?
2. On considère maintenant un tableau contenant une suite décroissante d'entiers : laquelle des trois méthodes de tri est la plus rapide ? Laquelle est la moins rapide ?

Exercice 5.3

Soient $s1$ et $s2$ deux suites de (respectivement) $n1$ et $n2$ nombres réels triées dans l'ordre croissant et rangées dans deux tableaux `t1` et `t2`. Écrire une fonction pour fusionner ces deux suites en une troisième également triée et la placer dans un tableau `t` (on supposera que la taille de `t` est suffisante).

Exercice 5.4

Un tri est dit `stable` s'il preserve l'ordre relatif des données consecutives identiques (par exemple : si une suite de noms d'étudiants est triée en fonction d'une note, un tri stable produit une suite dans laquelle les etudiants ayant la même note sont encore en ordre alphabetique). Lesquels de nos trois méthodes de tri sont stables ? (faire tourner les algorithmes sur un tableau de structures composées d'un entier et d'une chaîne de caractères)

Exercice 5.5

Une méthode de tri applicable à une situation particulière : le tri par dénombrement.
Il s'agit de trier une suite de n nombres entiers dont les valeurs sont comprises entre 0 et k.
Si k n'est pas très grand on peut utiliser l'algorithme suivant :

- Soit `s` le tableau contenant la suite, on calcule (à l'aide d'un tableau auxiliare `compte` de taille $k + 1$), la fréquence de chaque entier dans `s`.
- On modifie `compte` de façon à avoir dans `compte[i]` le nombre d'éléments de `s` ayant une valeur inférieure ou égale à i.
- On utilise le contenu de `compte` pour construire la suite triée dans un deuxième tableau auxiliaire `t` en utilisant l'instruction suivante :

```
pour tout i allant de n à 1 par pas de -1 faire
     t[compte[s[i]]]= s[i];
     compte[s[i]]= compte[s[i]]-1;
finpour
```

- On recopie la suite triée dans le tableau `s`.

1. Écrire une fonction pour calculer la fréquence des éléments d'une suite de n entiers positifs ayant une valeur inférieure à k.

2. Écrire une fonction qui effectue le tri par dénombrement d'une suite de n entiers positifs ayant valeur inférieure à k.