
ALGO5 : Travaux dirigés, séance 3 Ecriture d’algorithmes à partir de description en langue naturelle

1 Deux algorithmes de tri

On donne le principe de deux algorithmes de tri, et on précise quelle variante on va utiliser. La donnée est un tableau de n éléments, indicés de 1 à n . Le type des éléments est quelconque mais muni d’un ordre total. Le résultat est le tableau contenant une permutation des mêmes éléments mais dans l’ordre croissant.

Le travail consiste à :

- Lire et comprendre les descriptions de ces algorithmes en s’aidant de schémas.
- Ecrire le schéma algorithmique de ces deux tris.
- Etudier la complexité de ces algorithmes en termes de nombres de comparaisons : on s’intéressera tout d’abord aux cas $n = 2$ et $n = 3$, puis au cas général.

1.1 Tri par insertion

L’algorithme consiste en une itération pour $i = 2$ à n . A chaque pas de l’itération on insère à sa place l’élément d’indice i ; plus précisément, lors de la $i^{\text{ème}}$ itération les $i - 1^{\text{èmes}}$ premiers éléments sont triés et on insère le $i^{\text{ème}}$ en bonne place dans cette séquence. Initialement, l’élément 1 forme une séquence triée et à la fin les n éléments sont triés.

L’insertion est réalisée après une recherche séquentielle de l’emplacement k de l’élément i , puis un décalage vers la droite des éléments d’indices k à $i - 1$. L’algorithme classique effectue ces deux opérations ensemble, c’est-à-dire décale l’élément i vers la gauche (par un échange) jusqu’à ce qu’il atteigne sa “bonne” place.

1.2 Tri par sélection (du minimum)

L’algorithme consiste en une itération de $i = 1$ à $n - 1$. A chaque pas de l’itération on cherche le plus petit élément dans la partie non encore traitée et on le met à sa place. A la fin, les $n - 1$ premiers éléments sont à leur place, donc le dernier est bien placé aussi.

On effectue la sélection du minimum par un parcours des éléments de i à n en mémorisant son indice k , puis on effectue l’échange entre les éléments d’indice i et k .

Écriture d'algorithmes

2 Drapeau hollandais

On dispose d'un tableau T de n éléments, chaque élément est coloré avec une des 3 couleurs bleu, blanc ou rouge. L'objectif est de réorganiser le tableau de manière à ce que les éléments bleus soient sur la partie gauche du tableau, les éléments blancs au centre et les rouges en fin de tableau.

L'objectif est de réaliser ce réarrangement en utilisant un minimum de mémoire supplémentaire ; on s'autorise à avoir un peu de mémoire supplémentaire pour réaliser des échanges et gérer quelques indices.

L'algorithme gère trois indices compris entre 1 et n : i_1 , i_2 et i_3 . L'algorithme est une itération bâtie autour de la propriété suivante (invariant) : T contient une permutation des éléments du tableau initial et les éléments de 1 à $i_1 - 1$, de $i_2 + 1$ à i_3 et de $i_3 + 1$ à n sont de couleurs respectives C1, C2 et C3. L'élément courant à traiter est $T[i_1]$, il s'agit en fonction de sa couleur (couleur($T[i_1]$)) de le placer dans la bonne zone en jouant sur les valeurs des indices et des échanges éventuels entre éléments du tableau.

Questions :

1. Faire un dessin représentant la propriété invariante.
2. Déterminer des conditions initiales pour les indices i_1 , i_2 et i_3 .
3. Déterminer comment maintenir la propriété invariante selon la couleur du $i^{\text{ème}}$ élément.
4. Déterminer la condition permettant l'arrêt lorsque tous les éléments ont été traités.
5. Vous avez l'algorithme complet . . .
6. Ecrire un algorithme qui résout le même problème mais avec deux couleurs seulement.
7. Ecrire un programme de tri récursif qui s'appuie sur une décomposition en trois zones.