

Chapitre 5 : Tris

1 Présentation

Le tri est une opération qui vise à réordonner les éléments d'un tableau selon un ordre défini à l'avance. Cette relation d'ordre est définie sur une paire d'éléments et indique lequel se place avant l'autre.

Dans cette séance nous chercherons à ordonner les éléments d'un tableau d'entiers selon l'ordre croissant des entiers naturels, mais toute relation d'ordre entre deux éléments permet d'ordonner une liste de ces éléments.

Exemple: Un tableau non-trié:

7	3	9	5	1
---	---	---	---	---

Le *même* tableau (*ie* contenant les *mêmes* valeurs) trié:

1	3	5	7	9
---	---	---	---	---

Pour remplir cet objectif, de nombreux algorithmes ont été proposés. Ils ne sont pas tous équivalents en terme d'efficacité et peuvent notamment demander plus ou moins d'opérations élémentaires et donc être plus ou moins longs à exécuter pour une taille donnée de problème (la taille du tableau ou le nombre d'éléments à trier).

Nous chercherons lors de cette séance à comparer différents algorithmes classiques par des calculs empiriques de complexité. Nous étudierons notamment comment évolue le temps de calcul ainsi que le nombre d'opérations nécessaires pour ordonner le tableau en fonction de la taille du tableau.

Nous nous attendons à ce que le nombre d'opérations comme le temps de calcul augmente lorsque la taille du tableau (et donc le nombre d'éléments à trier) augmente. Cependant, nous mettrons en évidence que le nombre d'opérations et le temps de calcul augmentent plus ou moins vite selon les algorithmes.

2 Organisation de la séance

Dans un premier temps les 4 algorithmes à implémenter vous seront rapidement présentés par votre enseignant. Dans un second temps, 4 groupes ou plus d'étudiants seront formés. Chaque groupe prendra en charge l'étude et la rédaction d'un algorithme de tri. Chaque algo devra être traité par au moins un groupe. Lors de la phase préparatoire (30min), afin de progresser dans l'analyse et la conception des algorithmes, les étudiants utilisent des cartes pour simuler les éléments à trier. Les étudiants présentent enfin aux autres groupes leur algo en utilisant les cartes et en détaillant le code. Dans un dernier temps, chaque étudiant code les 4 algorithmes et mène

l'analyse en complexité détaillée dans la partie exercice.

3 Les algorithmes de tri

3.1 Tri par sélection du minimum

Sur un tableau de n éléments (numérotés de 1 à n), le principe du tri par sélection est le suivant :

- rechercher le plus petit élément du tableau, et l'échanger avec l'élément d'indice 0 ;
- rechercher le second plus petit élément du tableau, et l'échanger avec l'élément d'indice 1 ;
- etc... continuer de cette façon jusqu'à ce que le tableau soit entièrement trié.

3.2 Tri par insertion

Dans l'algorithme, on parcourt le tableau à trier du début à la fin. Au moment où on considère le i -ème élément, les éléments qui le précèdent sont déjà triés. Pour faire l'analogie avec l'exemple du jeu de cartes, lorsqu'on est à la i -ème étape du parcours, le i -ème élément est la carte saisie, les éléments précédents sont la main triée et les éléments suivants correspondent aux cartes encore mélangées sur la table.

L'objectif d'une étape est d'insérer le i -ème élément à sa place parmi ceux qui précèdent. Il faut pour cela trouver où l'élément doit être inséré en le comparant aux autres, puis décaler les éléments afin de pouvoir effectuer l'insertion. En pratique, ces deux actions sont fréquemment effectuées en une passe, qui consiste à faire « remonter » l'élément au fur et à mesure jusqu'à rencontrer un élément plus petit.

3.3 Tri à bulle

L'algorithme parcourt le tableau et compare les éléments consécutifs. Lorsque deux éléments consécutifs ne sont pas dans l'ordre, ils sont échangés.

Après un premier parcours complet du tableau, le plus grand élément est forcément en fin de tableau, à sa position définitive. En effet, aussitôt que le plus grand élément est rencontré durant le parcours, il est mal trié par rapport à tous les éléments suivants, donc échangé à chaque fois jusqu'à la fin du parcours (il remonte telle une bulle en haut du tableau).

Après le premier parcours, le plus grand élément étant à sa position définitive, il n'a plus à être traité. Le reste du tableau est en revanche encore en désordre. Il faut donc le parcourir à nouveau, en s'arrêtant à l'avant-dernier élément. Après ce deuxième parcours, les deux plus grands éléments sont à leur position définitive. Il faut donc répéter les parcours du tableau, jusqu'à ce que les deux plus petits éléments soient placés à leur position définitive.

3.4 Tri comptage

On suppose qu'on dispose d'un tableau `tab` composé de 100 entiers entre 0 et 30 (bornes comprises). Le procédé du tri par comptage est le suivant : on compte le nombre de « 0 », le nombre de « 1 », ..., le nombre de « 30 » présents dans `tab`, et on reconstruit `tab` en y ajoutant les valeurs selon leur quantité croissante (on ne trie pas les valeurs mais le comptage de ces valeurs au sein du tableau).