Informatique XI
Une introduction aux algorithmes de tri

Laurent Kaczmarek

PCSI^2 2013-2014
Lycée Louis Le Grand

Lundi 27 janvier 2014

Informatique XI
Une introduction aux algorithmes de tri

Laurent Kaczmarek

Les algorithmes de tri

Le tri par sélection

Le tri par insertion

Exercices

Principe

- L'objectif est, étant donnée une liste d'entiers, de trier ses éléments. Nous allons envisager différentes méthodes de tri et déterminer, pour chacune d'entre elles, leur complexité en nombre de comparaisons. Ces algorithmes se généralisent sans difficultés à des listes où l'ensemble des *clés* peut-être muni d'un ordre total.
- Un algorithme de tri s'effectue *en place* c'est la liste initiale qui est modifiée au fur et à mesure de l'algorithme jusqu'à être triée.
- Nous ne présenterons que les algorithmes. Leur programmation en Python sera effectuée lors des séances de TP.

Informatique XI
Une introduction aux algorithmes de tri
Laurent Kaczmarek
Les algorithmes de tri
Le tri par sélection
Le tri par insertion
Exercices

Le tri par sélection

Algorithme

- Le *tri par sélection* est une des méthodes de tri les plus simples.
- On commence par rechercher le plus petit élément de la liste que l'on va mettre à sa place en l'échangeant avec le premier élément.
- Puis on recommence avec le deuxième élément le plus petit, etc. . .
- La liste est ainsi divisée en deux parties : une première partie déjà triée et une deuxième partie contenant les éléments restant à trier.
- La liste est triée *en place*.

Algorithme

Voici l'algorithme :

Algorithme 1 : Tri par sélection

pour *0* $\leqslant$ *i* $\leqslant$ *n-1* **faire**
    i0 $\leftarrow$ la position du minimum de `t[i:len(t)]`;
    Échanger `t[i]` et `t[i0]`;
Renvoyer `t`;

Un exemple

- En triant la liste

  `t=[10,1,5,19,3,3]`

  on obtient successivement :
- `[1,10,5,19,3,3]`
- `[1,3,5,19,10,3]`
- `[1,3,3,19,10,5]`
- `[1,3,3,5,19,10]`
- `[1,3,3,5,10,19]`

Complexité

- Le nombre de comparaisons effectuées par le tri par sélection ne dépend que de la longueur de la liste en entrée. Ainsi, les complexités dans le pire et dans le meilleur des cas sont identiques.
- Le calcul du minimum de la sous-liste requiert $n - i - 1$ comparaisons.
- Le nombre total de comparaisons effectuées est donc

$$\sum_{i=0}^{n-1}(n-i-1) = \sum_{i=0}^{n-1} i = \frac{(n-1)n}{2}$$

- Dans le meilleur et dans le pire des cas, la complexité du tri par sélection est quadratique.

Informatique XI
Une introduction aux algorithmes de tri
Laurent Kaczmarek
Les algorithmes de tri
Le tri par sélection
Le tri par insertion
Exercices

Principe

- Le *tri par insertion* est généralement le tri utilisé pour classer des documents ou des cartes à jouer.
- On commence par prendre le premier élément à trier que l'on place en première position.
- Puis on garde une partie de la liste triée dans laquelle on insère les éléments restants à leur place un par un.
- La liste est triée *en place*.

Informatique XI
Une introduction aux algorithmes de tri

Laurent Kaczmarek

Les algorithmes de tri
Le tri par sélection
Le tri par insertion
Exercices

Le tri par sélection

Algorithme

Voici l'algorithme :

Algorithme 2 : Tri par insertion

pour *1* $\leqslant i \leqslant$ *n-1* **faire**
    Chercher $j \in [\![0, i-1]\!]$ tel que `t[j]` $\leqslant$ `t[i]` $<$ `t[j+1]`;
    Insérer `t[i]` entre `t[j]` et `t[j+1]`;
Renvoyer `t`;

Un exemple

- Pour `t=[10,1,5,19,3,3]`, on obtient :
- `[10,1,5,19,3,3]`
- `[1,10, 5,19,3,3]`
- `[1,5,10,19,3,3]`
- `[1,5,10,19,3,3]`
- `[1,3,5,10,19,3]`
- `[1,3,3,5,10,19]`

Complexité

La complexité de l'algorithme est entièrement conditionnée par le nombre de comparaisons nécessaires pour trouver l'indice d'insertion du nouvel élément. Nous allons décrire ci-dessous ce nombre de comparaisons dans le meilleur et dans le pire des cas et décrire ces cas critiques.

Complexité dans le meilleur des cas

Lorsque l'élément `t[i]` est déjà en place, il suffit d'une comparaison pour le constater (alors `t[i-1]` $\leqslant$ `t[i]`). L'algorithme effectue ainsi 1 comparaison par itération de la boucle. La complexité dans le meilleur des cas est ainsi linéaire. Les listes correspondant à ce cas sont les listes déjà triées.

Informatique XI
Une introduction aux algorithmes de tri
Laurent Kaczmarek
Les algorithmes de tri
Le tri par sélection
Le tri par insertion
Exercices

Complexité dans le pire des cas

- Le nombre maximal de comparaisons à chaque itération de la boucle est égal à i.
- Alors, `t[i]` $\leqslant$ `t[0]`. L'algorithme effectue ainsi i comparaisons lors de la i-ème itération.
- Le nombre total de comparaisons est ainsi égal à $(n-1)n/2$.
- La complexité dans le pire des cas est ainsi quadratique.
- Les listes correspondant à ce cas sont les listes triées en ordre inverse.

Informatique XI
Une introduction aux algorithmes de tri
Laurent Kaczmarek
Les algorithmes de tri
Le tri par sélection
Le tri par insertion
Exercices

Exercice 1 : le tri-bulle

- Le principe du tri à bulle est le suivant : parcourir le tableau en comparant les couples d'éléments successifs. Lorsque deux éléments successifs ne sont pas dans l'ordre croissant, ils sont échangés : le maximum arrive en position finale en remontant comme une bulle de champagne. On recommence alors l'opération jusqu'à obtenir une liste entièrement triée.
- En déduire un algorithme de tri.

Exercice 2 : le tri par dénombrement

- Soit N un entier naturel. On suppose que les listes sont constituées d'entiers compris entre 0 et N. Écrire une fonction `triPostier(t)` qui trie une liste `t` en temps linéaire par rapport à sa longueur.
- On pourra utiliser une liste auxiliaire `tcomptage` qui compte les occurrences de chaque entier de $[\![0, N]\!]$.