

Cours 3: Tri

1 Introduction

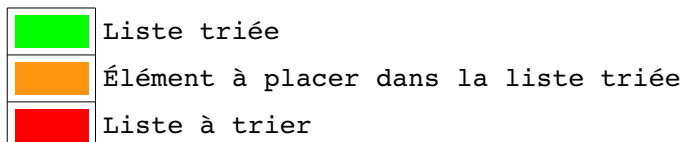
Définition: On désigne par "tri" l'opération consistant à ordonner un ensemble d'éléments en fonction de clés sur lesquelles est définie une relation d'ordre.

L'étude du tri est également intéressante en elle-même car il s'agit sans doute du domaine de l'algorithmique qui a été le plus étudié et qui a conduit à des résultats remarquables sur la construction d'algorithmes et l'étude de leur complexité.

Il existe plusieurs algorithmes de tri, nous allons étudier les plus courants.

2 Le tri par insertion

La liste à trier est divisée en deux : la partie gauche est triée, et la partie droite est non-triée. A chaque passage, le premier élément de la partie non-triée est inséré dans la partie triée, de telle sorte que cette dernière reste triée. Pour une liste de n éléments, $n - 1$ passages sont nécessaires.



1^{er} passage:



2^{ème} passage:



3^{ème} passage:



4^{ème} passage:



...

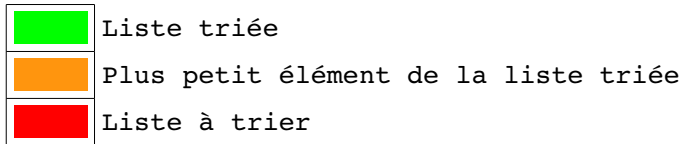
9^{ème} passage:



Cette méthode peut nécessiter de décaler les éléments dans la liste triée, ce qui peut être coûteux. Cet algorithme peut être intéressant pour classer des valeurs en temps réel (au fur et à mesure qu'elles arrivent). La complexité est $O(n^2)$ en moyenne et dans le pire des cas.

3 Le tri par sélection

Comme pour le tri par insertion, la liste à trier est divisée en deux : la partie gauche est triée, et la partie droite est non-triée. Cette fois-ci, tous les éléments de la liste triée sont inférieurs à ceux de la liste non triée. Le principe du tri par sélection est d'aller chercher le plus petit élément de la liste non triée pour le permuter avec le premier élément de cette même liste.



1^{er} passage:



2^{ème} passage:



3^{ème} passage:



4^{ème} passage:



...


9^{ème} passage:



Cette méthode ne nécessite pas de décaler les éléments, mais requiert une recherche du plus petit élément de la liste. La complexité est $O(n^2)$ en moyenne et dans le pire des cas.

4 Le tri à bulles

Le principe du tri à bulles est de comparer deux à deux les éléments x_n et x_{n+1} consécutifs d'un tableau et d'effectuer une permutation si $x_{n+1} < x_n$. On continue de trier jusqu'à ce qu'il y ait un passage complet sans permutation (la liste est entièrement triée).

 Élément à permuter

1^{er} passage:

10	5	7	3	1	2	4	9	12	5
5	10	7	3	1	2	4	9	12	5
5	7	10	3	1	2	4	9	12	5
5	7	3	10	1	2	4	9	12	5
5	7	3	1	10	2	4	9	12	5
5	7	3	1	2	10	4	9	12	5
5	7	3	1	2	4	10	9	12	5
5	7	3	1	2	4	9	10	12	5

2^{ème} passage:

5	7	3	1	2	4	9	10	5	12
5	3	7	1	2	4	9	10	5	12
5	3	1	7	2	4	9	10	5	12
5	3	1	2	7	4	9	10	5	12
5	3	1	2	4	7	9	10	5	12

3^{ème} passage:

5	3	1	2	4	7	9	5	10	12
3	5	1	2	4	7	9	5	10	12
...

Le tri à bulle est particulièrement efficace sur des données déjà triées. La complexité est $O(n^2)$ en moyenne et dans le pire des cas.

5 Tri rapide (Quick Sort)

A chaque étape, l'algorithme sélectionne un élément particulier du tableau : le pivot. Le tableau est alors réorganisé en trois parties : les éléments plus petits que le pivot (à gauche), le pivot, et les éléments plus grands que le pivot (à droite). Le pivot est donc placé à sa position finale. La réorganisation peut être réalisée uniquement avec des permutations. On recommence ensuite, récursivement, pour la partie gauche et la partie droite du tableau.

	Pivot
	Éléments à permuter

1^{er} appel:

10	5	7	3	1	2	4	9	12	5
10	5	7	3	1	2	4	9	5	12
5	5	7	3	1	2	4	9	10	12

2^{ème} appel:

5	5	7	3	1	2	4	9
5	5	4	3	1	2	7	9
5	5	4	3	1	2	7	9
2	5	4	3	1	5	7	9

3^{ème} appel:

2	5	4	3	1
2	1	4	3	5
2	1	4	3	5
1	2	4	3	5

4^{ème} appel:

1

5^{ème} appel:

4	3	5
4	3	5
3	4	5

...

La complexité est $O(n \cdot \log_2(n))$ en moyenne et $O(n^2)$ dans le pire des cas.