CGM-MAT3
ALGOSTAT

Rapport d'analyse des algorithmes

Partie I : Présentation des algorithmes

Le tri à bulle :

Le principe du tri à bulle est d'abord de parcourir tout le tableau, à chaque paire il intervertit les éléments. (Exemple voir 5 et 1) il va les intervertir. Il va faire les échanges à chaque tours jusqu'à que le tableau soit trié.

Il est efficace quand le tableau est en parti trié et quand pour le tableau composé d'éléments random. Par contre il montre des lacunes lors d'un tri avec un tableau inversé.

Le tri par sélection :

Le principe du tri à sélection est le suivant : il recherche le plus petit élément du tableau et l'échange avec le premier élément. Il procède de cette façon jusqu'à que le tableau soit entièrement trié.

Il est plutôt efficace avec toutes sortes de "petit" tableau. Puisqu'il doit à chaque fois parcourir tout le tableau, il n'est pas optimisé pour les tableaux comprenant un grand nombre d'entrée.

Le tri par insertion :

Ce tri parcourt le tableau et, pour chaque élément rencontré, recherche la position à lui affecter. Cette position est définie comme étant juste avant le premier élément rencontré qui soit supérieur à celui en cours. Une fois cette position trouvée, tous les nombres se trouvant entre la position initiale de l'élément et sa position finale sont déplacés pour pouvoir enfin écrire l'élément à sa place.

Il est efficace avec les tableaux en parties triés et ceux avec des valeurs similaire, A l'inverse il n'est pas efficace avec les tableaux inverse.

Le tri à peigne :

Le principe du trie à peigne est de comparer des éléments plus lointains, puis de raccourcir progressivement l'intervalle entre les éléments pour aboutir finalement à un tri à bulle classique.

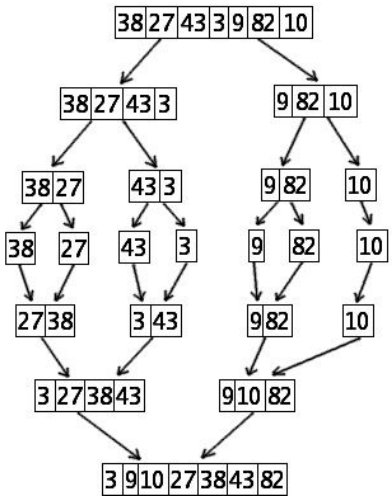
Il a une efficacité plutôt semblable sur la plupart des tableaux, cependant il montre des difficulté avec les grande valeurs et les tableaux random.

Le tri de Shell :

Le principe du tri de Shell est qu'il trie chaque liste d'éléments séparés de n positions chacun avec le Tri insertion . Se tri effectue plusieurs fois cette opération en diminuant n jusqu'à n=1 ce qui équivaut à trier toutes les valeurs ensemble.

Il est efficace avec le tri partiellement trié, par contre il n'est pas optimisé avec les séries random.

Le tri fusion :



Le principe de ce tri repose sur la division du tableau à trier en deux. En effet, le tri fusion coupe le tableau en deux sous tableaux plus ou moins égaux (comprenant plus ou moins le même nombre de case). Il va répéter cette opération jusqu'à ce que le tableau ne soit plus divisible en deux. Un fois ce but atteint, il va comparer les deux valeurs et va les intervertir si la valeur A est plus petite que la valeur B. Puis, il va merger le tableau trié avec sa "moitié" sur lequel le tri aura aussi été réalisé. C'est lors du merge qu'il associe les deux tableaux et qu'il effectue le tri sur les deux. Il effectue cette opération de manière récursive jusqu'à récupérer le tableau initial, trié.

Le tri fusion est optimisé pour les "petits" tableaux : puisqu'il découpe chaque tableau en sous tableau de manière récursive, il peut facilement devenir obsolète si la mémoire allouée n'est pas suffisante !

Le tri rapide :

C'est l'algorithme le plus efficace. Pour effectuer ce tri, il faut sélectionner un "pivot" (c'est souvent le premier élément du tableau) et on coupe le tableau en deux : toutes les valeurs inférieures au pivot sont classées dans un tableau et toutes les plus grandes dans un autre. On réitère l'opération jusqu'à ce que les sous-tableaux soient triés et que l'on ai mergé le tout dans le tableau initial.

Ce tri est particulièrement peu efficace sur la série de tableau déjà trié.

Partie II : présentation des séries de test

Série “tableau déjà trié” :

Exemple de tableau déjà trié :

0	1	2	3	4	5	6	7	8	9

Le tri rapide n’est pas recommandé pour le tableau déjà trié, peu importe le nom d’élément. En effet, il serait plus lent car il découpe en tableau de deux éléments à tester à chaque fois et cela lui prend beaucoup de temps et d’itération pour rien.

Série “tableau trié inverse” :

Exemple de tableau trié inverse:

9	8	7	6	5	4	3	2	1	0

Le tri rapide est le plus optimisé pour cette série. Il permet de ranger efficacement les éléments en effectuant un minimum d’itération.

Série “tableau random” :

Exemple de tableau random :

59	10	21	54	4	9	52	37	874	154

Le tri à insertion et le tri de Shell sont les plus pertinents pour la série random. Par contre le tri à bulles et l tri par sélection ne sont pas du tout optimisés.
Le tri à insertion ne va pas parcourir le tableau 9 fois dans ce cas si comme le ferait le tri à bulle. Il est donc plus optimisé.

Série “tableau quasiment trié” :

Exemple de série déjà trié :

0	1	2	3	4	9	5	6	7	2

Le tri à peigne et le tri à insertion sont les plus rapides pour ce cas. Par contre le tri à bulles et à sélections sont les moins optimisés.

Le tri à peigne va quasiment trié le tableau du premier coup en effet, il va faire comme le tri à bulle mais en plus rapide car il prend des petite sélection.

Série “tableau avec beaucoup de doublons” :

Exemple de série déjà trié :

5	5	8	8	6	6	6	0	1	2

Le tri par insertion et le tri à shell sont normalement les plus efficaces.
Les tris rapide et à bulle sont les moins optimisés pour ce cas. Dans le cas du tri rapide, il risquerait de tomber trop souvent sur deux valeurs identiques, ce qui, comme dans le cas du tableau déjà trié, le ralenti.

Partie III : comparatif entre théorie et résultats

Pour 10 éléments :

Nom du tri		Déjà trié	Trié inverse	Quasiment trié	Beaucoup de doublons	Random
Insertion	Attendu	25	50	25	25	25
	Réel	9	54	9	11	42
Sélection	Attendu	45	45	45	45	45
	Réel	65	65	65	65	65
Bulle	Attendu	10	100	100	100	100
	Réel	65	65	65	65	65
Peigne	Attendu	10	10	10	10	10
	Réel	37	37	37	37	37
Shell	Attendu	10	10	10	10	10
	Réel	25	38	25	29	40
Fusion	Attendu	10	10	10	10	10
	Réel	68	68	68	68	68
Rapide	Attendu	10	10	10	10	10
	Réel	45	45	45	30	23

Pour 100 éléments :

Nom du tri		Déjà trié	Trie inverse	Quasiment trié	Beaucoup de doublons	Random
Insertion	Attendu	10 000	10 000	10 000	10 000	10 000
	Réel	99	99	200	2085	
Sélection	Attendu	10 000	10 000	10 000	10 000	10 000
	Réel	5150	5150	5150	5150	5150
Bulle	Attendu	10 000	10 000	10 000	10 000	10 000
	Réel	5150	5150	5150	5150	5150
Peigne	Attendu	200	200	200	200	200
	Réel	1016	1016	1016	1016	1016
Shell	Attendu	200	200	200	200	200
	Réel	515	515	552	652	837
Fusion	Attendu	200	200	200	200	200
	Réel	31 470	31 470	30 980	14 005	10 400
Rapide	Attendu	200	200	200	200	200
	Réel	5 150	3 0126	29 636	12 661	9 056

Pour 1 000 éléments :

Nom du tri		Déjà trié	Trie inverse	Quasiment trié	Beaucoup de doublons	Random
Insertion	Attendu	1 000 000	1 000 000	1 000 000	1 000 000	1 000 000
	Réel	501 500	500 499	10 740	195 472	252 699
Sélection	Attendu	1 000 000	1 000 000	1 000 000	1 000 000	1 000 000
	Réel	501 500	501 500	501 500	501 500	501 500
Bulle	Attendu	1 000 000	1 000 000	1 000 000	1 000 000	1 000 000
	Réel	501 500	501 500	501 500	501 500	501 500
Peigne	Attendu	3 000	3 000	3 000	3 000	3 000
	Réel	18 735	18 735	18 735	18 735	18 735
Shell	Attendu	3 000	3 000	3 000	3 000	3 000
	Réel	8 093	11 317	8 964	10 410	14 805
Fusion	Attendu	3 000	3 000	3 000	3 000	3 000
	Réel	19 952	19 952	19 952	19 952	19 952
Rapide	Attendu	3 000	3 000	3 000	3 000	3 000
	Réel	499 500	499 500	481 433	112 665	1 0428

Pour 10 000 éléments :

Nom du tri		Déjà trié	Trie inverse	Quasiment trié	Beaucoup de doublons	Random
Insertion	Attendu	100 000 000	100 000 000	100 000 000	100 000 000	100 000 000
	Réel	9999	5 004 999	1 017 729	19 786 463	25 225 476
Sélection	Attendu	100 000 000	100 000 000	100 000 000	100 000 000	100 000 000
	Réel	50 015 000	50 015 000	50 015 000	50 015 000	50 015 000
Bulle	Attendu	100 000 000	100 000 000	100 000 000	100 000 000	100 000 000
	Réel	50 015 000	50 015 000	50 015 000	50 015 000	50 015 000
Peigne	Attendu	40 000	40 000	40 000	40 000	40 000
	Réel	276 770	276 770	276 770	276 770	276 770
Shell	Attendu	40 000	40 000	40 000	40 000	40 000
	Réel	110 845	153 769	126 285	140 589	210 212
Fusion	Attendu	40 000	40 000	40 000	40 000	40 000
	Réel	267 232	267 232	267 232	267 232	267 232
Rapide	Attendu	40 000	40 000	40 000	40 000	40 000
	Réel	49 995 000	49 995 000	48 022 442	10 730 496	159 158

Vu les résultats actuelles entrés dans le tableau, il est fortement possible que nous n’ayons pas totalement compris comment calculer le résultat attendu et/ou le résultat réel.