

Structure des données et algorithmes

Projet 1

KAMANGU Samuel (s180093)

Mars 2024

Exercice 1

Point 1

Les résultats repris ci-dessous sont des moyennes calculées à partir de 10 tests réalisés pour chaque algorithme.

Type de tableau	Aléatoire				
Algo	1000	5000	10000	50000	100000
0	0,002383	0,027133	0,099236	2,436068	9,843686
1	0,000161	0,000582	0,00131	0,007586	0,016422
2	0,000098	0,000385	0,000879	0,004862	0,010319
3	0,001675	0,025467	0,101534	2,539272	10,075317
4	0,00016	0,000726	0,001567	0,009215	0,019622
5	0,00014	0,000721	0,001562	0,010599	0,02447
6	0,002382	0,055062	0,23208	6,300212	26,191836

Calcul empirique des temps d'exécution sur base de tableaux aléatoires

Type de tableau	Croissant				
Algo	1000	5000	10000	50000	100000
0	0,000011	0,000048	0,000095	0,000431	0,000763
1	0,000161	0,00067	0,001311	0,005267	0,009129
2	0,000062	0,000272	0,000535	0,001872	0,003386
3	0,002298	0,027198	0,102504	2,515355	10,068484
4	0,00019	0,000605	0,001304	0,00718	0,015119
5	0,00003	0,00011	0,000222	0,001315	0,002798
6	0,000005	0,000013	0,000025	0,000115	0,000232

Calcul empirique des temps d'exécution sur base de tableaux croissants

Type de tableau	Décroissant				
Algo	1000	5000	10000	50000	100000
0	0,004044	0,051181	0,196718	4,884923	19,807579
1	0,000076	0,000297	0,000619	0,003658	0,007839
2	0,000051	0,00021	0,000467	0,00282	0,006023
3	0,001518	0,027042	0,107498	2,737718	10,970817
4	0,000123	0,000579	0,001244	0,007005	0,014692
5	0,000045	0,000209	0,00045	0,002558	0,005488
6	0,002173	0,044859	0,179672	4,590988	18,677686

Calcul empirique des temps d'exécution sur base de tableaux décroissants

Type de tableau	Aléatoire				
Algo	1000	5000	10000	50000	100000
0	251290	6283320	25062611	627358102	2499224891
1	8712	55180	120390	717978	1536326
2	10583	69646	150152	889829	1821018
3	499500	12497500	49995000	1249975000	4999950000
4	16941	107709	235346	1409882	3019555
5	14094	105016	229658	1669393	4020949
6	494247	12487770	49978529	1249959069	4999723872

Calcul empirique du nombre de comparaisons sur base de tableaux aléatoires

Type de tableau	Croissant				
Algo	1000	5000	10000	50000	100000
0	999	4999	9999	49999	99999
1	5044	32004	69008	401952	853904
2	9451	57963	125916	736177	1572338
3	499500	12497500	49995000	1249975000	4999950000
4	17583	112126	244460	1455438	3112517
5	5457	35084	75243	455719	967146
6	999	4999	9999	49999	99999

Calcul empirique du nombre de comparaisons sur base de tableaux croissants

Type de tableau	Décroissant				
Algo	1000	5000	10000	50000	100000
0	499500	12497500	49995000	1249975000	4999950000
1	4932	29804	64608	382512	815024
2	13135	87123	193981	1210382	2620334
3	499500	12497500	49995000	1249975000	4999950000
4	15965	103227	226682	1366047	2926640
5	8550	51491	120190	689218	1533494
6	499500	12497500	49995000	1249975000	4999950000

Calcul empirique du nombre de comparaisons sur base de tableaux décroissants

Type de tableau	Aléatoire				
Algo	1000	5000	10000	50000	100000
0	250293	6278329	25052623	627308117	2499124912
1	993	4991	9987	49989	99982
2	3173	18424	39106	223286	473277
3	999	4999	9999	49999	99999
4	9122	57138	124215	737388	1575028
5	9085	72070	158784	1236588	3095098
6	250293	6278329	25052623	627308117	2499124912

Calcul empirique du nombre de swaps sur base de tableaux croissants

Type de tableau	Croissant				
Algo	1000	5000	10000	50000	100000
0	0	0	0	0	0
1	0	0	0	0	0
2	976	4094	8190	34464	68928
3	999	4999	9999	49999	99999
4	9708	60932	131956	773304	1650854
5	0	0	0	0	0
6	0	0	0	0	0

Calcul empirique du nombre de swaps sur base de tableaux croissants

Type de tableau	Décroissant				
Algo	1000	5000	10000	50000	100000
0	499500	12497500	49995000	1249975000	4999950000
1	500	2500	5000	25000	50000
2	2467	11666	23376	124095	248243
3	999	4999	9999	49999	99999
4	8316	53436	116696	698892	1497434
5	3920	19212	53704	270768	619654
6	499500	12497500	49995000	1249975000	4999950000

Calcul empirique du nombre de swaps sur base de tableaux croissants

Point 2

Voici les résultats d'identification des algorithmes présumés :

Algo	Complexités			stable?	en place?	algorithme présumé
	meilleur cas	pire cas	cas moyen			
0	$O(n)$	$O(n^2)$	$O(n^2)$	oui	oui	insertion sort
1	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	oui	non	merge sort
2	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	non	oui	quick sort
3	$O(n^2)$	$O(n^2)$	$O(n^2)$	non	oui	selection sort
4	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	non	oui	heap sort
5	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$	oui	oui	inconnu
6	$O(n)$	$O(n^2)$	$O(n^2)$	oui	oui	bubble sort

Identification des algorithmes présumés

Les résultats des premiers tests permettent d'identifier que les algorithmes de tri 2, 3 et 4 ne sont pas stables et correspondent donc aux algorithmes quick sort, tri par sélection et tri par tas.

L'algorithme de tri par sélection a été identifié car les temps d'exécution évoluent de manière quadratique par rapport aux tailles des tableaux dans les meilleurs cas, les pires cas et les cas moyens (aléatoire). Ce qui rejoint les complexités théoriques.

Pour les mêmes raisons, on peut identifier que les algorithmes 0 et 6 sont les algorithmes de tri par insertion et tri à bulles. Pour identifier lequel correspond à quel algorithme, j'ai réalisé des tests supplémentaires. Ces tests consistaient à échanger 2 éléments d'un tableau croissant.

Par élimination, nous savons que les algorithmes 1 et 5 sont l'algorithme de tri par fusion et l'algorithme inconnu.

En observant les temps d'exécution des algorithmes 2 et 4. On observe que l'algo 4 est plus constant peu importe l'ordre des éléments. Ceci caractérise l'algorithme de tri par tas. Par élimination, on peut déterminer que l'algo 2 est le quick sort.

Swap des éléments 0 et 1 :

L'échange de ces 2 éléments du tableau a renvoyé comme résultat que les algorithmes 0, 1, 5 et 6 n'ont réalisé qu'un seul swap. Il ne pouvaient s'agir que des algos : tri par insertion, tri à bulles et tri par fusion.

Par contre, le nombre de comparaisons de l'algorithme 6 correspond au double de celui de l'algorithme 0. C'est pourquoi je présume que l'algorithme 6 est le tri à bulles car par définition, il doit réaliser des comparaisons de l'ensemble du tableau après chaque swap (une fois de plus que celui par insertion donc 2 fois plus car il n'y a eu qu'un seul swap).

Algorithme 0

Average number of comparisons: 9999

Average number of swaps: 1

Algorithm 6

Average number of comparisons: 19997

Average number of swaps: 1

Swap des éléments 0 et N-1 :

L'échange de ces 2 éléments du tableau a renvoyé comme résultat qu'uniquement l'algorithme 1 n'a réalisé qu'un seul swap. Cet élément m'a permis d'identifier l'algorithme de tri par fusion.

Par élimination, l'algorithme 5 est l'algorithme inconnu.