Algorithmics	Student information	Date	Number of session
	UO: UO269412		2
	Surname: Carrillo		
	Name: Javier		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Time measurements for sorting algorithms

Insertion

Insertion			
n	**sorted(10^-5 seconds)**	**inverse(ms)**	**random(ms)**
10000	21	72	31
20000	53	316	123
40000	130	428	238
80000	268	1552	943
160000	497	6339	2850
320000	1009	25003	11297
640000	2215	99061	44161
1280000	4669	394583	173764

Let's calculate the theorical values of a time t2, taking into account different n.

Starting with the best case, which is the sorted one, we know that the complexity of the method is O(n); hence we calculate t2 using the formula t2 = n2/n1 * t1

For n1 = 10000, t1 = 21, n2 = 20000, we calculate that tht2 = 42, close to the obtt2 = 53

For n1 = 80000, t1 = 130, n2 = 160000, we calculate that tht2 = 260, close to the obtt2 = 268.

Then, we have the worst case, the inverse one. We know that the complexity of the method is O(n^2); hence we calculate t2 using the formula t2 = n2^2/n1^2 * t1

For n1 = 10000, t1 = 72, n2 = 20000, we calculate that tht2 = 288, close to the obtt2 = 316

For n1 = 80000, t1 = 1552, n2 = 160000, we calculate that tht2 = 6208, close to the obtt2 = 6339.

Finally, we operate with an average case, being the random one. We know that the complexity of the method is O(n^2); hence we calculate t2 using the formula t2 = n2^2/n1^2 * t1

For n1 = 10000, t1 = 31, n2 = 20000, we calculate that tht2 = 124, close to the obtt2 = 123

For n1 = 160000, t1 = 2850, n2 = 320000, we calculate that tht2 = 11400, close to the obtt2 = 11297.

Algorithmics	Student information	Date	Number of session
	UO: UO269412		2
	Surname: Carrillo		
	Name: Javier		

Selection

Selection			
n	**sorted(ms)**	**inverse(ms)**	**random(ms)**
10000	18	49	17
20000	54	166	50
40000	214	818	266
80000	973	1473	1072
160000	3087	10248	3017
320000	11919	41093	12746
640000	48062	157137	48951
1280000	204553	633892	216189

Let's calculate the theorical values of a time t2, taking into account different n.

All three cases of selection share the same complexity, being O(n^2) so the times for such cases are calculated the same, using the formula t2 = n2^2/n1^2 * t1

Starting with the best case, which is the sorted one:

For n1 = 10000, t1 = 18, n2 = 20000, we calculate that tht2 = 72, close to the obtt2 = 54

For n1 = 160000, t1 = 973, n2 = 320000, we calculate that tht2 = 12348, close to the obtt2 = 11919.

Then, we have the worst case, the inverse one:

For n1 = 10000, t1 = 49, n2 = 20000, we calculate that tht2 = 196, close to the obtt2 = 166

For n1 = 160000, t1 = 10248, n2 = 320000, we calculate that tht2 = 40992, close to the obtt2 = 41093.

Finally, we operate with an average case, being the random one:

For n1 = 10000, t1 = 17, n2 = 20000, we calculate that tht2 = 69, close to the obtt2 = 50

For n1 = 160000, t1 = 3017, n2 = 320000, we calculate that tht2 = 12068, close to the obtt2 = 12746.

Algorithmics	Student information	Date	Number of session
	UO: UO269412		2
	Surname: Carrillo		
	Name: Javier		

Bubble

Bubble			
n	**sorted(ms)**	**inverse(ms)**	**random(ms)**
10000	43	95	128
20000	206	346	576
40000	362	731	2753
80000	2692	5306	10562
160000	9898	21055	42030
320000	41039	84438	164782
640000	158257	339287	651870

Let's calculate the theorical values of a time t2, taking into account different n.

All three cases of selection share the same complexity, being O(n^2) so the times for such cases are calculated the same, using the formula t2 = n2^2/n1^2 * t1

Starting with the best case, which is the sorted one:

For n1 = 10000, t1 = 43, n2 = 20000, we calculate that tht2 = 184, close to the obtt2 = 206

For n1 = 80000, t1 = 2692, n2 = 160000, we calculate that tht2 = 10768, close to the obtt2 = 9898.

Then, we have the worst case, the inverse one:

For n1 = 10000, t1 = 95, n2 = 20000, we calculate that tht2 = 280, close to the obtt2 = 346

For n1 = 80000, t1 = 5306, n2 = 160000, we calculate that tht2 = 21224, close to the obtt2 = 21055.

Finally, we operate with an average case, being the random one:

For n1 = 10000, t1 = 128, n2 = 20000, we calculate that tht2 = 496, close to the obtt2 = 576

For n1 = 80000, t1 = 10562, n2 = 160000, we calculate that tht2 = 42248, close to the obtt2 = 42030.

Algorithmics	Student information	Date	Number of session
	UO: UO269412		2
	Surname: Carrillo		
	Name: Javier		

Quicksort (central element)

Quicksort			
n	**sorted(ms)**	**inverse(ms)**	**random(10^-1 seconds)**
10000	117	155	74
20000	227	259	145
40000	444	555	601
80000	861	1187	2169
160000	1864	2436	
320000	3917	4974	
640000	8087	10804	

Let's calculate the theorical values of a time t2, taking into account different n.

In these samples taken, we will consider that the complexity is O(nlogn) and as such, we will use the formula t2 = n2/n1 * log(n2)/log(n1) / t1

Starting with the sorted samples:

For n1 = 10000, t1 = 117, n2 = 20000, we calculate that tht2 = 251, close to the obtt2 = 227

For n1 = 80000, t1 = 861, n2 = 160000, we calculate that tht2 = 1827, close to the obtt2 = 1864.

Then, we have the inverse case

For n1 = 10000, t1 = 155, n2 = 20000, we calculate that tht2 = 333, close to the obtt2 = 259

For n1 = 80000, t1 = 1187, n2 = 160000, we calculate that tht2 = 2519, close to the obtt2 = 2436.

Finally, we operate with the random one

For n1 = 10000, t1 = 74, n2 = 20000, we calculate that tht2 = 159, close to the obtt2 = 145