Laboratory work nr. 2
Analyzing of sorting algorithms

Bîrcu Maxim

November 12, 2015

Student:	Bîrcu Maxim
Instructor:	Cojanu Irina

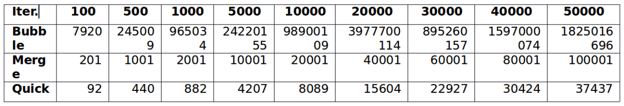
Work purpose

1. **Studing of 3 sorting algorithms**
2. **Implementation of those sorting algorithms**
3. **Results analyzing**

Results

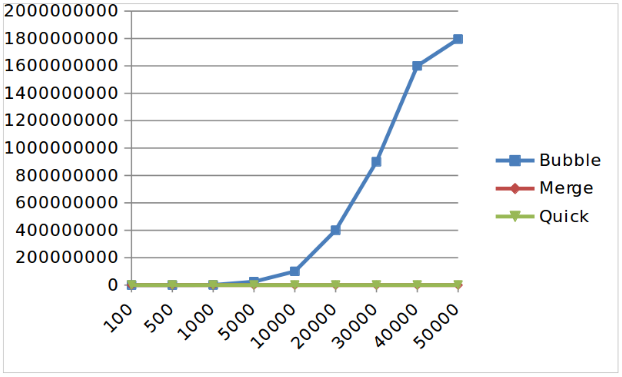
I've studied three methods of sorting (bubble sort, merge sort and quick sort) for different rows of names arranged in random order, and calculated the number of iterations for each of these methods to the same string, the results I have scored in a table.

1 Random inputs



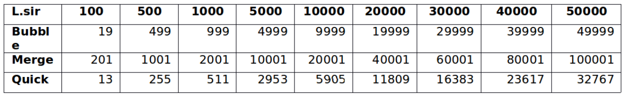
Iter.	**100**	**500**	**1000**	**5000**	**10000**	**20000**	**30000**	**40000**	**50000**
Bubble	7920	245009	965034	24220155	98900109	3977700114	895260157	1597000074	1825016696
Merge	201	1001	2001	10001	20001	40001	60001	80001	100001
Quick	92	440	882	4207	8089	15604	22927	30424	37437

For better understanding of the results I've placed them on a graph.



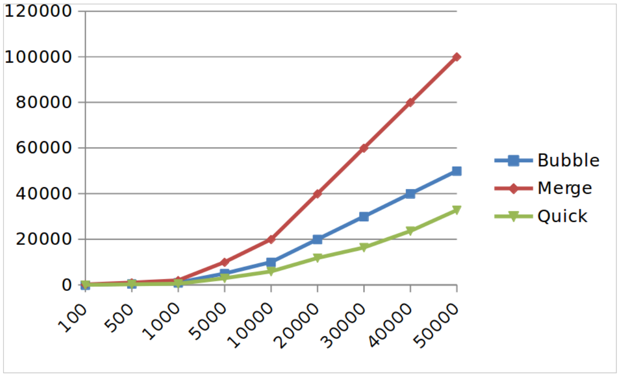
We denote that sorting method with most iterations is Bubble sort which was expected. And the increasing of the number of iterations for merge an quick sort are pretty the same. So the most inefficient algorithm is Bubble sort.

2 Favorable case



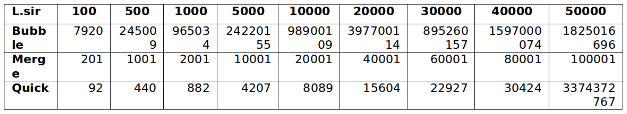
L.sir	100	500	1000	5000	10000	20000	30000	40000	50000
Bubble	19	499	999	4999	9999	19999	29999	39999	49999
Merge	201	1001	2001	10001	20001	40001	60001	80001	100001
Quick	13	255	511	2953	5905	11809	16383	23617	32767

For better understanding of the results I've placed them on a graph.



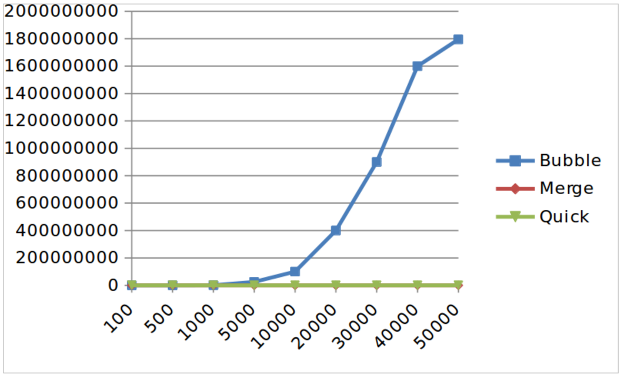
We can observe that in the favorable case all algorithms has uniform increasing , the faster increasing is for merge sort , and also we can observe that the number of iterations of merge sort algorithm do not depend of the case because in random case we have the same results for this algorithm. Also if we compare Quick and Bubble sort we can observe a small difference and make the conclusion that Quick sort is quite faster than Bubble and Merge sort.

3 Unfavorable case

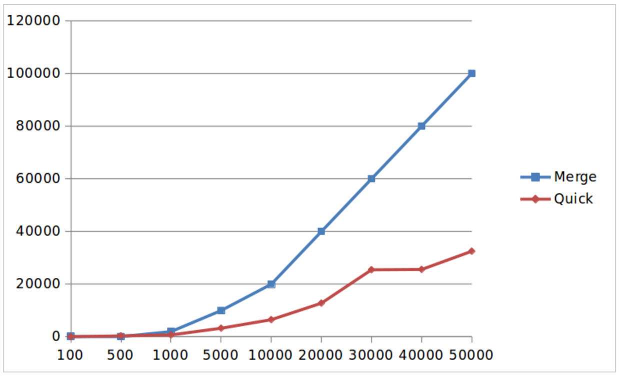


L.sir	100	500	1000	5000	10000	20000	30000	40000	50000
Bubble	7920	245009	965034	24220155	98900109	397700114	895260157	1597000074	1825016696
Merge	201	1001	2001	10001	20001	40001	60001	80001	100001
Quick	92	440	882	4207	8089	15604	22927	30424	3374372767

For better understanding of the results I've placed them on a graph.



Here we can observe from the graph that Bubble sort has an abnormal increasing and therefore it is inefficient and we should study the last 2 methods in another graph without bubble sort.



Now we can see the difference between the number of iterations of the merge sort an quick sort algorithms. We can observe that Merge sort has a uniform increasing of iterations number in comparison with Quick sort but Quick sort has a slower increasing than Merge sort.

4 Conclusion:

After all research and algorithms comparison I've observed that the efficiency of the algorithm in some cases depends on the input size and form which means the random case when the list is sorted randomly , the favorable case when the list is sorted ascendant and the unfavorable case when the list is sorted descendant. Also I observed that in all 3 cases the quick sort algorithm was the most efficient. About Merge an Bubble sort I can say that if we have to chose one of this 2 algorithms our choosing will depend on the problem. So if we will have to check if a list is sorted than the most efficient algorithm will be Bubble sort but if we should sort a descendant sorted list probably merge sort will be a better choosing than Bubble sort.