BBM204 Algortihms Lab. First Assignment Report

Furkan KIZILTAN
21607877
kiziltan.furkan@gmail.com

March 12, 2018

1 Problem Definition

In this assignment participants faced against one of the main problems of modern age in computing, efficency while sorting data. There are a lot of ways to solve a problem and sorting is no different. One can manage this task in multiple ways. But to decide which one is better, one needs to consider the time and the resources(memory space) it requires. To observe the performance varience between several algorithms participants need to implement three of them and record results accordingly.

2 Findings

My code contains three different algorithms which are Selection Sort, Bubble Sort and Quick Sort. Also I've been given five data sets which contains 100, 1000, 50000, 100000, All trafic traces respectively. Values measured according to the 48th feature can be found in the table below.

Table 1: Findings(i5-4590, 16GB DDR3)

Algorithms&Data Sets	TrafficFlow100	TrafficFlow1000	TrafficFlow50000	TrafficFlow100000	TrafficFlowAll
Selection Sort	368330ns	10ms	3514ms	8716ms	3.27min
Bubble Sort	662497ns	14ms	3885ms	9075ms	3.58min
Quick Sort	242657ns	1ms	48ms	82ms	146ms

3 Discussion

Selection Sort's and Bubble Sort's time complexities are $O(n^2)$ since they compare the whole array in each pass therefore has two nested loops. However, that is not the case for Quick Sort. In Quick Sort "divide and conquer" approach is used. Thanks to this approach data is divided into parts in each iteration, allowing us to reduce the comperisons to $O(n \log n)$, where N is the number of items in the list. As can be seen from result obtained, $O(n^2)$ seems considerably demanding compared to $O(n \log n)$ as it rises exponentially. As for memory usage, those algorithms do not demand extra space in order to run. They constantly change the current array/list by swapping elements when necessary. That being said, there are other algorithms that needs extra space like Merge Sort also.