

*Marmara University*

***Faculty of Engineering***

***Computer Science Engineering***

***Analysis Of Algoritm CSE2046***

HW3 Raport;

Harun Büyüktepe 150115020

We designed an experiment to compare different algorithms for selection problem. Our implementation solves problem of finding k’th smallest element in an unsorted list of n numbers. Firstly we implement ordinary ;sorting by Insertion-sort and returning the k’th element in the list, sorting by Merge-sort and returning the k’th element in the list, not sorting the list, but applying quick select algorithm, which is based on array partitioning and applying quick select algorithm with using median-of-three selection.

In this experiment, we compared these four methods for different values of k and various input lists of various sizes and various characteristics. We considered to work on sample inputs from small to largest to make decisions best clarify the performance characteristics of algorithms. Also we used physical unit of time, count actual number of basic operation's executions.

We prepared an array builder for the assignment, This array constructor creates an array with random values from 0 to the maximum integer size. The array is called in a ‘for loop’ with frequency of 25.In this for loop, In this loop, we invoke the elements in the different indexes on the array with in ascending order such first element,median of array and last element.

We know insertion sort have simple quadratic,  algorithms such as selection sort or bubble sort.Also we saw experimental analysis of insertion sort have quadratic time complexity.

We know merge sort algorithm with average time complexity and generally worst-case complexity O(n log n) .Merge sort stays always stable.Also we measure parallel measurement.

We didn’t sort the array so time come so messy .We estimate this outcome. Median value always come the biggest because program scan more than other index values.

We know median of three’s time complexity in avarage case O(n log n) and worst case  . We got values close to the values which we were expecting.

We have found some results.For example,if we want to sort an array with efficient time complexity and less comparison ,we can use merge sort;if we find element from beginning or end in very big data with efficient time complexity and less comparison, we can use quick select algoritm.However, if we have small array size ,we can use insertion sort because of basic implementation reason.If we want to find median of array , it would be wise to use the median of three method.

Finally we got that different techniques work efficiently in different situations and conditions. The programer decides which method to use with considering number of comparison,time complexity and memory usage to make efficient program.