

**T.C.**

**MARMARA UNIVERSITY**

**FACULTY of ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

**CSE 2046 Analysis of Algorithms**

Title of the Project

Finding k’th Smallest Element with Various Sort Technique

Group Members

150119861 – Ahmet BOZBAY

150119860 – Onur KAYA

**SORTS**

Contents

[1. Insertion sort 3](#_Toc103202033)

[2. Merge Sort 4](#_Toc103202034)

[3. Quick Sort 5](#_Toc103202035)

[4. Partial Selection Sort 7](#_Toc103202036)

[5. Partial Heap Sort 10](#_Toc103202037)

[6. Analysis Part 13](#_Toc103202038)

[a. Time 13](#_Toc103202039)

[b. Analysing Process Time of Algorithms 14](#_Toc103202040)

[Insertionsortdescendingorder 1 3](#_Toc103201952)

[InsertionSortFigure 2 InsertionSortFigure 3 4](#_Toc103201953)

[MergeSortDividingPart 4 4](#_Toc103201954)

[MergeSortCorrectOrder 5 5](#_Toc103201955)

[MergeSortFigure 6 MergeSortFigure 7 5](#_Toc103201956)

[QuickSortPivot 8 6](#_Toc103201957)

[QuickSortReplacingPivot 9 7](#_Toc103201958)

[QuickSortFigure 10 QuickSortFigure 11 7](#_Toc103201959)

[PartialSelectionSortOrdering 12 8](#_Toc103201960)

[PartialSelectionSortCorrectPosition13 8](#_Toc103201961)

[SelectionSortFigure14 9](#_Toc103201962)

[PartialHeapSortFindMaxHeap 15 10](#_Toc103201963)

[PartailHeapSortBuildHeap 16 11](#_Toc103201964)

[HeapSortFigure 17 12](#_Toc103201965)

[ISPprocesstime1 13](#_Toc103202100)

[MSPProcesstime2 13](#_Toc103202101)

[QSPProcesstime 3 13](#_Toc103202102)

[PSSProcesstime 4 13](#_Toc103202103)

[PHSProcesstime5 13](#_Toc103202104)

[QuickselectsortProcesstime 6 13](#_Toc103202105)

# Insertion sort

Insertion sort is a sorting algorithm that puts each term to be added in its order. this order goes from smallest to largest, just like we do when counting numbers. here, if the number we choose is greater than the first number we have, it is added to the right, if it is smaller, it is added to the left, however, the same process is applied to all the numbers to be added. metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Insertionsortdescendingorder 1

For example, suppose we have a number array. In this array, our first element is automatically sorted. Secondly, if the number we are going to add is greater than the first number, it is added to the right of this number, if it is less, it is added to the left of this number and updated as the first number in the array. continue this process and all other numbers are inserted.

ok içeren bir resim

Açıklama otomatik olarak oluşturuldu

InsertionSortFigure 2 InsertionSortFigure 3

# Merge Sort

Merge sort is a process in which problems are divided into certain parts and finally solved by combining them all, and the divide and conquer technique is applied. In this technique, we first divide the sequences we have into two parts,

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

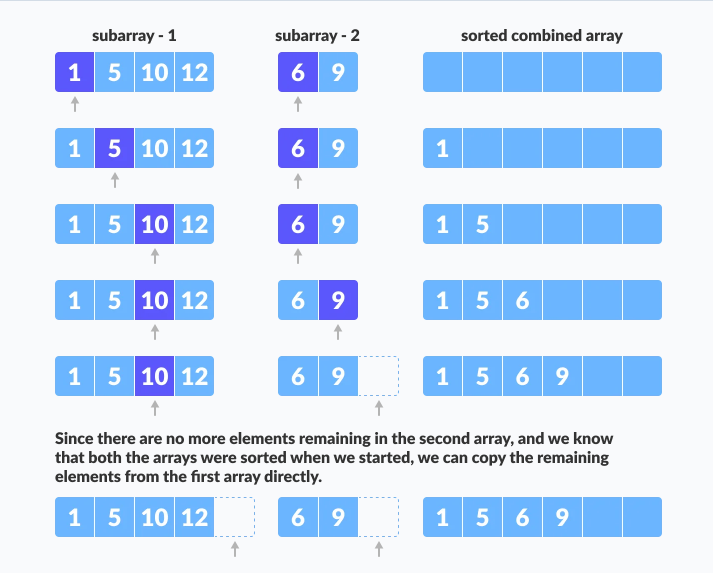
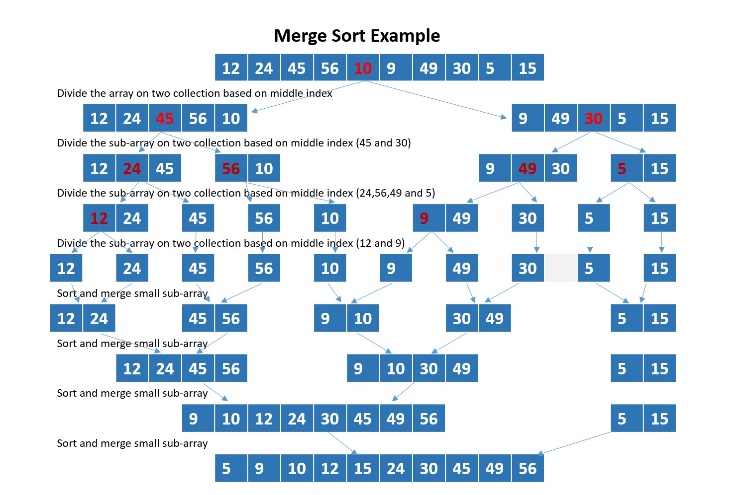
MergeSortDividingPart 4

to the left and right of the middle point. If we cannot reach the desired order, we divide these sub-series into two again.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

MergeSortCorrectOrder 5

At the end of this process, when we get the order we want, we combine all these sub-arrays.

MergeSortFigure 6 MergeSortFigure 7

# Quick Sort

Quicksort is an algorithm in which we divide it into sub-arrays by choosing a pivot element like the divide and conquer operation.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

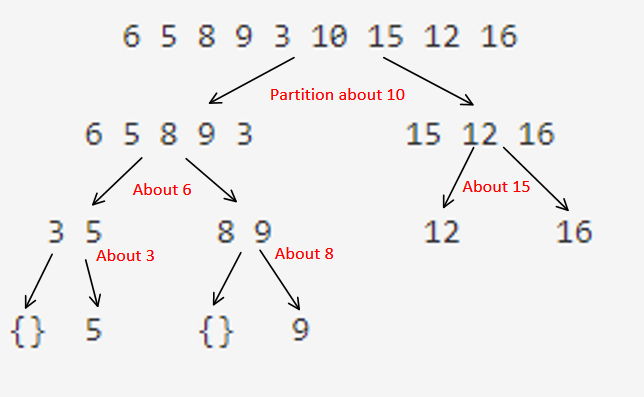
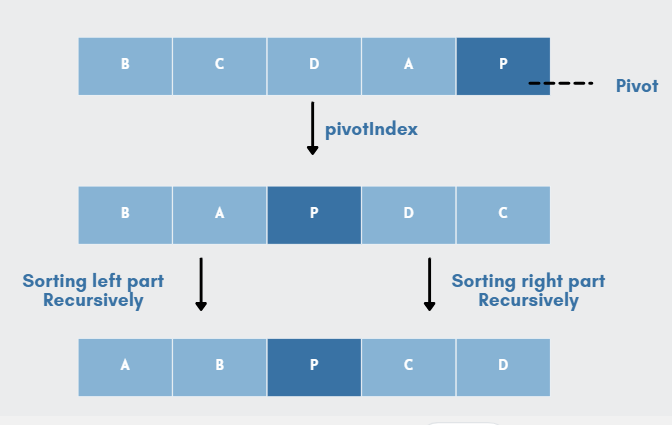
QuickSortPivot 8

Here the pivot element is chosen as the rightmost element of the array. After this element is selected, it is rearranged by ordering the smaller ones from the pivot to the left and the larger ones to the right. After this operation, if the element we selected is larger than the pivot A second element is selected, and the pivot is compared again. If the pivot is greater than the second element, the second selected element is replaced with the first selected element. Then, the next larger number is determined, and the process is repeated with a smaller element. This process is terminated when the second last element is found. Finally, the pivot replaced by the second element specified.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

QuickSortReplacingPivot 9



QuickSortFigure 10 QuickSortFigure 11

# Partial Selection Sort

Partial selection sort algorithm is a selection algorithm that is used to select the minimum (or maximum) item in the selection order.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

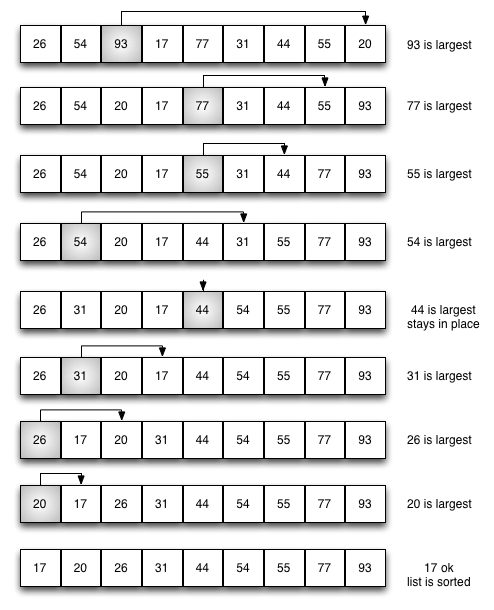
PartialSelectionSortOrdering 12

Besides, in the case of k=1, it is more preferred than other sorting. In summary, this selection algorithm is an algorithm that takes a shorter time compared to the others, but it is more inefficient than the others. But when the value of k is small, its efficiency increases considerably. In this system, the smallest value is taken to the beginning and the process continues until the desired number of elements is accumulated and the number in the row of the relevant element is returned.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

PartialSelectionSortCorrectPosition13



SelectionSortFigure14

# Partial Heap Sort

Partial heapsort is an algorithm by which heaps are sorted. Here the heap is used for fast sorting, an unsorted array is chosen as the maximum heap so that the largest item is our first item in the array.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

PartialHeapSortFindMaxHeap 15

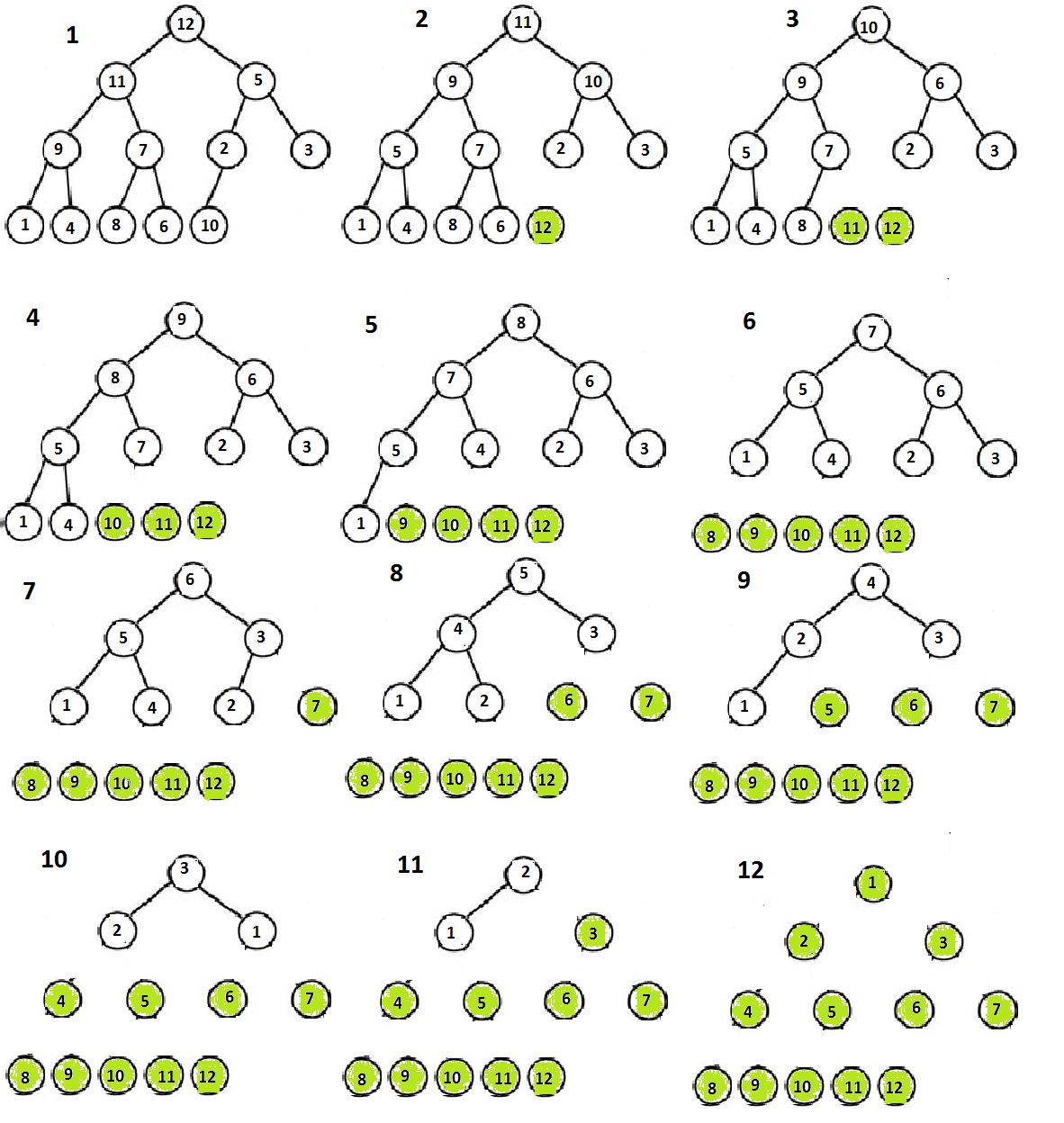
We will have the smallest value in the array. After that, the scrolling operation is done and the second largest item is updated as the new root of the array, after this action, our largest value in the array is ignored and fixed as the maximum value of the array, so as not to leave its place. After this operation, the first and last item is again their locations change and the stack is rearranged to reach the maximum value. This process is repeated until it reaches

the root node, and finally a completely sorted array is obtained. This algorithm is divided into three as worst, average and best in performance. not a stable ranking system.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

PartailHeapSortBuildHeap 16



HeapSortFigure 17

# Analysis Part

## Time

Comparison of Sorting Process Time of Algorithms



ISPprocesstime1



MSPProcesstime2



QSPProcesstime 3



PSSProcesstime 4



PHSProcesstime5



QuickselectsortProcesstime 6

## Analysing Process Time of Algorithms

This experiment was done by randomly creating a list of numbers between 1 and 100000 3000 times.

According to this experiment, we can see that;

Merge Sort Process and Partial Heap Sort are almost close to each other and have been the fastest sorting technique.

After these, respectively;

Partial Selection Sort, Insertion Sort and Quick sort Process are the sort techniques that can perform the fastest process operations, respectively.

In addition, the quick select algorithm technique does not even take time on the processor side with the selection it makes.