

CS435DE - Lab 5

Saurab Ghimire

417555

Problem 1: Solution

QuickSort Steps for Array [1, 6, 2, 4, 3, 5] using Leftmost Pivot:

Step 1:

Pivot = Left element = 1.

After partitions, we get: [1] + [6, 2, 4, 3, 5]

Step 2:

Array = [6, 2, 4, 3, 5],

Pivot = 6.

After partitions, we get: [2, 4, 3, 5] + [6]

Step 3:

Array = [2, 4, 3, 5],

Pivot = 2.

After partitions, we get: [2] + [4, 3, 5]

Step 4:

Array = [4, 3, 5],

Pivot = 4.

After partitions, we get: [3] + [4] + [5]

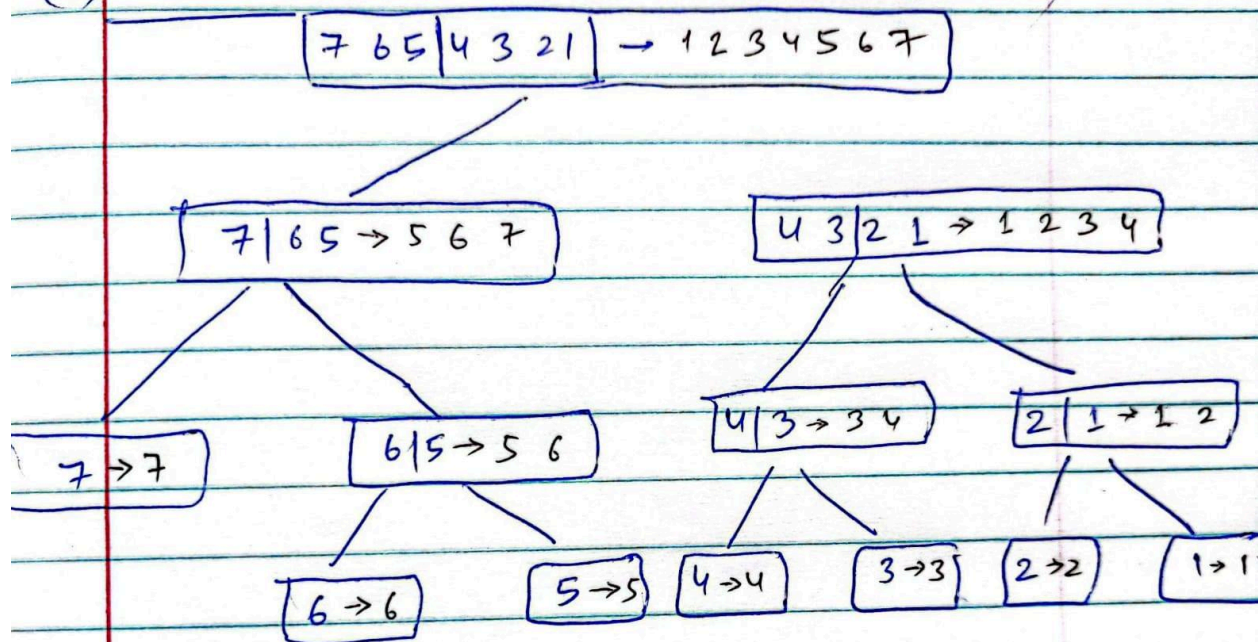
Step 5:

Finally, the array is sorted: [1, 2, 3, 4, 5, 6].

Problem 2: Solution

Saurab Ghimire (617555)

(2) Solution:



Pseudo code for mergeSortPlus

```
1  Algorithm MergeSortPlus(A)
2  Input: Array A of n integers
3  Output: Sorted array A
4
5  if  $n \leq 20$  then
6      InsertionSort(A)
7      return
8
9   $mid \leftarrow n / 2$ 
10  $left \leftarrow A[0 : mid-1]$ 
11  $right \leftarrow A[mid : n-1]$ 
12
13 MergeSortPlus(left)
14 MergeSortPlus(right)
15
16  $A \leftarrow merge(left, right)$ 
17
18 Algorithm InsertionSort(A)
19 Input: Array A of size n
20 Output: Sorted array A
21
22 for  $i \leftarrow 1$  to  $n-1$  do
23      $key \leftarrow A[i]$ 
24      $j \leftarrow i - 1$ 
25     while  $j \geq 0$  and  $A[j] > key$  do
26          $A[j + 1] \leftarrow A[j]$ 
27          $j \leftarrow j - 1$ 
28      $A[j + 1] \leftarrow key$ 
29
```

Using the subroutines in the given code, we can implement the code for mergesort plus where if $right - left + 1 \leq 20$, we use insertion sort.

```

1 package lab4;
2 import sortroutines.InsertionSort;
3
4 public class MergeSortPlus {
5     InsertionSort insertionSort = new InsertionSort();
6     public void mergeSortPlus(int[] arr, int left, int right) {
7         if (right - left + 1 <= 20) {
8             insertionSort.sort(arr);
9             return;
10        }
11
12        int mid = left + (right - left) / 2;
13
14        mergeSortPlus(arr, left, mid);
15        mergeSortPlus(arr, left: mid + 1, right);
16
17        mergeSortPlus(arr, left, right);
18    }
19 }

```

To verify the execution time between mergeSort and mergeSortPlus, I generated array of large elements and recorded execution times. With mergeSortPlus having insertionSort for more than 20, it was faster for smaller arrays, but for large inputs, performance was similar. The results are not conclusive.

Problem 4: Solution