Practical-12(F-26)

Problem Statement:

Assume we have two input and two output tapes to perform the sorting. The internal memory can hold and sort m records at a time. Write a program in java for external sorting. Find out time complexity.

Code:

```
import java.util.Scanner;
class MergeSort
{
 void merge(int arr[], int p, int q, int r)
   {
    int n1 = q - p + 1;
    int n2 = r - q;
    int L[] = new int[n1];
    int M[] = new int[n2];

    for (int i = 0; i < n1; i++)
      L[i] = arr[p + i];
    for (int j = 0; j < n2; j++)
      M[j] = arr[q + 1 + j];
    int i, j, k;
    i = 0;
    j = 0;
    k = p;
    while (i < n1 && j < n2)
    {
      if (L[i] <= M[j])
      {
        arr[k] = L[i];
        i++;
      } else
      {
        arr[k] = M[j];
        j++;
      }
      k++;
    }
    while (i < n1)
    {
      arr[k] = L[i];
      i++;
      k++;
    }

    while (j < n2)
    {
      arr[k] = M[j];
      j++;
      k++;
    }
  }


  void mergeSort(int arr[], int l, int r)
   {
     if (l < r)
     {
      int m = (l + r) / 2;

      mergeSort(arr, l, m);
      mergeSort(arr, m + 1, r);
      merge(arr, l, m, r);
    }
  }
  static void printArray(int arr[])
  {
    int n = arr.length;
    for (int i = 0; i < n; ++i)
      System.out.print(arr[i] + " ");
    System.out.println();
  }

  public static void main(String args[])
  {
  Scanner sc = new Scanner(System.in);
// Take the array size from the user
System.out.println("Enter the size of the array: ");
int arr_size = 0;
if (sc.hasNextInt()) {
arr_size = sc.nextInt();
}
int[] arr = new int[arr_size];
System.out.println(
"Enter the elements of the array: ");
for (int i = 0; i < arr_size; i++) {
if (sc.hasNextInt()) {
arr[i] = sc.nextInt();
}
}
    //int arr[] = { 6, 5, 12, 10, 9, 1 };

    MergeSort ob = new MergeSort();
    ob.mergeSort(arr, 0, arr_size - 1);
    System.out.println("Sorted array:");
    printArray(arr);
  }
}
```

OUTPUT :-

```
Enter the size of the array:
4
Enter the elements of the array:
10
9
8
7
Sorted array:
7 8 9 10
```