SORTING - II

Merge two sorted arrays

Given two arrays A and B, containing N and M elements both sorted in ascending order, create a new array C containing N+M elements in sorted order.

```
Input :

3 4
2 5 9
1 3 4 12

Output :

1 2 3 4 5 9 12
```

Solution Code

```
i = 0; // Initial index of first subarray
j = 0; // Initial index of second subarray
k = 0; // Initial index of merged subarray
while (i < n1 && j < n2)
{
    if (A[i] <= B[j])
    {
        C[k] = A[i];
        i++;
    }
    else
    {
        C[k] = B[j];
        j++;
    }
    k++;
}
```

```
/* Copy the remaining elements of L[], if there
       are any */
    while (i < n1)
    {
        C[k] = A[i];
        i++;
        k++;
    }

/* Copy the remaining elements of R[], if there
       are any */
    while (j < n2)
    {
        C[k] = B[j];
        j++;
        k++;
    }
```

Merge Sort

Extension of the previous problem, works by using Divide and Conquer.

```
MergeSort(arr[], l,  r)
If r > l
     1. Find the middle point to divide the array into two halves:
             middle m = (l+r)/2
     2. Call mergeSort for first half:
             Call mergeSort(arr, l, m)
     3. Call mergeSort for second half:
             Call mergeSort(arr, m+1, r)
     4. Merge the two halves sorted in step 2 and 3:
             Call merge(arr, l, m, r)
```

Merge Sort Code

```
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        // Same as (l+r)/2, but avoids overflow for
        // large l and h
        int m = l+(r-l)/2;

        // Sort first and second halves
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}
```

Efficiency of Merge Sort

1. Recursive relation :

   T(n) = T(n/2) + O(n)

2. Time complexity = O(N logN)

3. Memory Complexity = O(n)

Counting Inversions

Given an array A, find the number of “inversions”.

Formally, an “inversion” means a pair (i,j) such that i < j but A[i] > A[j].

```
Input :
5
2 4 1 3 5

Output :
3
```

In-Built Sort Function

Uses a combination of Quick Sort, Heap Sort and Insertion Sort to sort an array in O(N logN) time.

Usage :

```
int arr[n];

sort( arr, arr + n );
```

Custom Comparator Function

```
bool compare(int a, int b)
{
     return a>b;
}


sort( arr, arr+n, compare );
```

String Confusion

https://www.hackerrank.com/contests/zephyr-bits-goa/challenges/the-string-confusion/problem

Arshia, Awani, Jahanvi and Sneha, are busy solving a problem. Since childhood, they were taught that a comes before b and c comes before d, and so on...but, today, they have encountered a problem given by Khyati, where she predefines her own order of alphabets (P) they are supposed to sort the alphabets in the given order by Khyati. There are multiple cases in file, each test case having a pattern(order of 26 characters) and then final string which they need to sort accordingly. Please help them!

Input Format

The first line contains T. T denoting the number of test cases.Each test case consists of two lines.The first line contains the pattern string P indicating the relative ordering of alphabets and the second line contains the final string F to be sorted.

Thank You !