**WEEK 4:**

**Question 1:** Given an unsorted array of integers, design an algorithm and implement it using a program to sort an array of elements by dividing the array into two subarrays and combining these subarrays after sorting each one of them. Your program should also find number of comparisons and inversions during sorting the array.

#include <iostream>

using namespace std;

int getInvCount(int arr[],int n);

void merge(int a[],int l,int m,int r,int &c);

void mergesort(int arr[],int l,int r,int &c)

{

if(l<r)

{

int m=(l+r)/2;

mergesort(arr,l,m,c);

mergesort(arr,m+1,r,c);

merge(arr,l,m,r,c);

}

}

void merge(int a[],int l,int m,int r,int &c)

{

int n1=m-l+1;

int n2=r-m;

int L[n1],R[n2];

for(int i=0;i<n1;i++)

L[i]=a[l+i];

for(int j=0;j<n2;j++)

R[j]=a[m+1+j];

int i=0,j=0,k=l;

while(i<n1 && j<n2)

{

if(L[i]<=R[j])

a[k++]=L[i++];

else

a[k++]=R[j++];

c++;

}

while(i<n1)

a[k++]=L[i++];

while(j<n2)

a[k++]=R[j++];

}

int getInvCount(int arr[],int n)

{

int inv=0;

for(int i=0;i<n-1;i++)

for(int j=i+1;j<n;j++)

if(arr[i]>arr[j])

inv++;

return inv;

}

int main()

{

int T;

cout<<"enter the number of test cases"<<endl;

cin>>T;

while(T--)

{

int inv=0;

int c=0;

int n;

cout<<"enter the number of array elements"<<endl;

cin>>n;

int a[n];

cout<<"enter the array elements"<<endl;

for(int i=0;i<n;i++)

cin>>a[i];

inv=getInvCount(a,n);

mergesort(a,0,n-1,c);

cout<<"sorted array:"<<endl;

for(int i=0;i<n;i++)

cout<<a[i]<<" ";

cout<<endl<<"Comparisons ="<<c<<endl;

cout<<"Inversions ="<<inv<<endl;

}

return 0;

}

**Output:**

enter the number of test cases

3

enter the number of array elements

8

enter the array elements

23 65 21 76 46 89 45 32

sorted array:

21 23 32 45 46 65 76 89

Comparisons =16

Inversions =13

enter the number of array elements

10

enter the array elements

54 65 34 76 78 97 46 32 51 21

sorted array:

21 32 34 46 51 54 65 76 78 97

Comparisons =22

Inversions =28

enter the number of array elements

15

enter the array elements

63 42 223 645 652 31 324 22 553 12 54 65 86 46 325

sorted array:

12 22 31 42 46 54 63 65 86 223 324 325 553 645 652

Comparisons =43

Inversions =54

Process returned 0 (0x0) execution time : 84.379 s

Press any key to continue.

**Question 2:** Given an unsorted array of integers, design an algorithm and implement it using a program to sort an array of elements by partitioning the array into two subarrays based on a pivot element such that one of the sub array holds values smaller than the pivot element while another sub array holds values greater than the pivot element. Pivot element should be selected randomly from the array. Your program should also find number of comparisons and swaps required for sorting the array.

#include <iostream>

using namespace std;

void swap(int \*a, int \*b);

int partition(int array[], int low, int high,int &c,int &s);

void quickSort(int array[], int low, int high,int &c,int &s) {

if (low < high) {

int pi = partition(array, low, high,c,s);

quickSort(array, low, pi - 1,c,s);

quickSort(array, pi + 1, high,c,s);

}

}

int partition(int array[], int low, int high,int &c,int &s) {

int pivot = array[high];

int i = (low - 1);

for (int j = low; j < high; j++) {

c++;

if (array[j] <= pivot) {

i++;

swap(&array[i], &array[j]);

s++;

}

}

swap(&array[i + 1], &array[high]);

s++;

return (i + 1);

}

void swap(int \*a, int \*b) {

int t = \*a;

\*a = \*b;

\*b = t;

}

int main()

{

int T;

cout<<"enter the number of test cases"<<endl;

cin>>T;

while(T--)

{

int s=0;

int c=0;

int n;

cout<<"enter the number of array elements"<<endl;

cin>>n;

int a[n];

cout<<"enter the array elements"<<endl;

for(int i=0;i<n;i++)

cin>>a[i];

quickSort(a,0,n-1,c,s);

cout<<"sorted array is:"<<endl;

for(int i=0;i<n;i++)

cout<<a[i]<<" ";

cout<<endl<<"Comparisons ="<<c<<endl;

cout<<"Swaps ="<<s<<endl;

}

return 0;

}

**Output:**

enter the number of test cases

3

enter the number of array elements

8

enter the array elements

23 65 21 76 46 89 45 32

sorted array is:

21 23 32 45 46 65 76 89

Comparisons =14

Swaps =10

enter the number of array elements

10

enter the array elements

54 65 34 76 78 97 46 32 51 21

sorted array is:

21 32 34 46 51 54 65 76 78 97

Comparisons =29

Swaps =21

enter the number of array elements

15

enter the array elements

63 42 223 645 652 31 324 22 553 12 54 65 86 46 325

sorted array is:

12 22 31 42 46 54 63 65 86 223 324 325 553 645 652

Comparisons =45

Swaps =39

Process returned 0 (0x0) execution time : 100.123 s

Press any key to continue.**Question 3:** Given an unsorted array of integers, design an algorithm and implement it using a program to find Kth smallest or largest element in the array. (Worst case Time Complexity = O(n))

#include<iostream>

using namespace std;

void merge(int arr[],int l,int mid,int h)

{

int count=0;

int i=l,j=mid+1;

int temp[h-l+1];

int k=0;

while (i<=mid && j<=h)

{

if (arr[i]<arr[j])

temp[k++]=arr[i++];

else

{

temp[k++]=arr[j++];

count+=mid-i+1;

}

}

for (;i<=mid;)

temp[k++]=arr[i++];

for (;j<=h;)

temp[k++]=arr[j++];

for (int f=0;f<k;f++)

arr[f+l]=temp[f];

}

void merge\_sort(int arr[],int l,int h)

{

if (l<h)

{

int mid=l+(h-l)/2;

merge\_sort(arr,l,mid);

merge\_sort(arr,mid+1,h);

merge(arr,l,mid,h);

}

}

int main()

{

int t;

cout<<"enter the number of test cases"<<endl;

cin>>t;

while (t--)

{

int n;

cout<<"enter the number of array elements"<<endl;

cin>>n;

int arr[n];

cout<<"enter the array elements"<<endl;

for (int i=0;i<n;i++)

cin>>arr[i];

int k;

cout<<"enter the key"<<endl;

cin>>k;

merge\_sort(arr,0,n-1);

int flag=0;

cout<<arr[k-1]<<endl;

}

}

**Output:**

enter the number of test cases

2

enter the number of array elements

10

enter the array elements

123 656 54 765 344 514 765 34 765 234

enter the key

3

123

enter the number of array elements

15

enter the array elements

43 64 13 78 864 346 786 456 21 19 8 434 76 270 601

enter the key

8

78

Process returned 0 (0x0) execution time : 87.981 s

Press any key to continue.