NAME: JIGYASU DEO

Univ. Roll No. : 2015245

SECTION : M.L

CLASS ROLL NO : 32

SUBJECT : DESIGN AND ANALYSIS OF ALGORITHM

SUBJECT CODE : TCS 505

Tutorial - 3

Ans 1-6 = Done in Assignment - 1

Ans 7 = Program to find two indexes such that A[i] + A[j] = K

```
int main()
{
    int n, key;
    bool flag = false;
    cin >> n;
    vector<int> v(n);
    for (int i=0; i<n; i++)
    {
        cin>> v[i];
    }
    cin >> key;
    map<int, int> mp;
    for (int i=0; i<n; i++)
    {
        int temp = key - v[i];
        if (mp.find(temp) == mp.end())
        {
            mp[v[i]] = i;
        }
        else
        {
            cout << i << " " << mp[x];
            flag = true;
        }   break;
    }
```

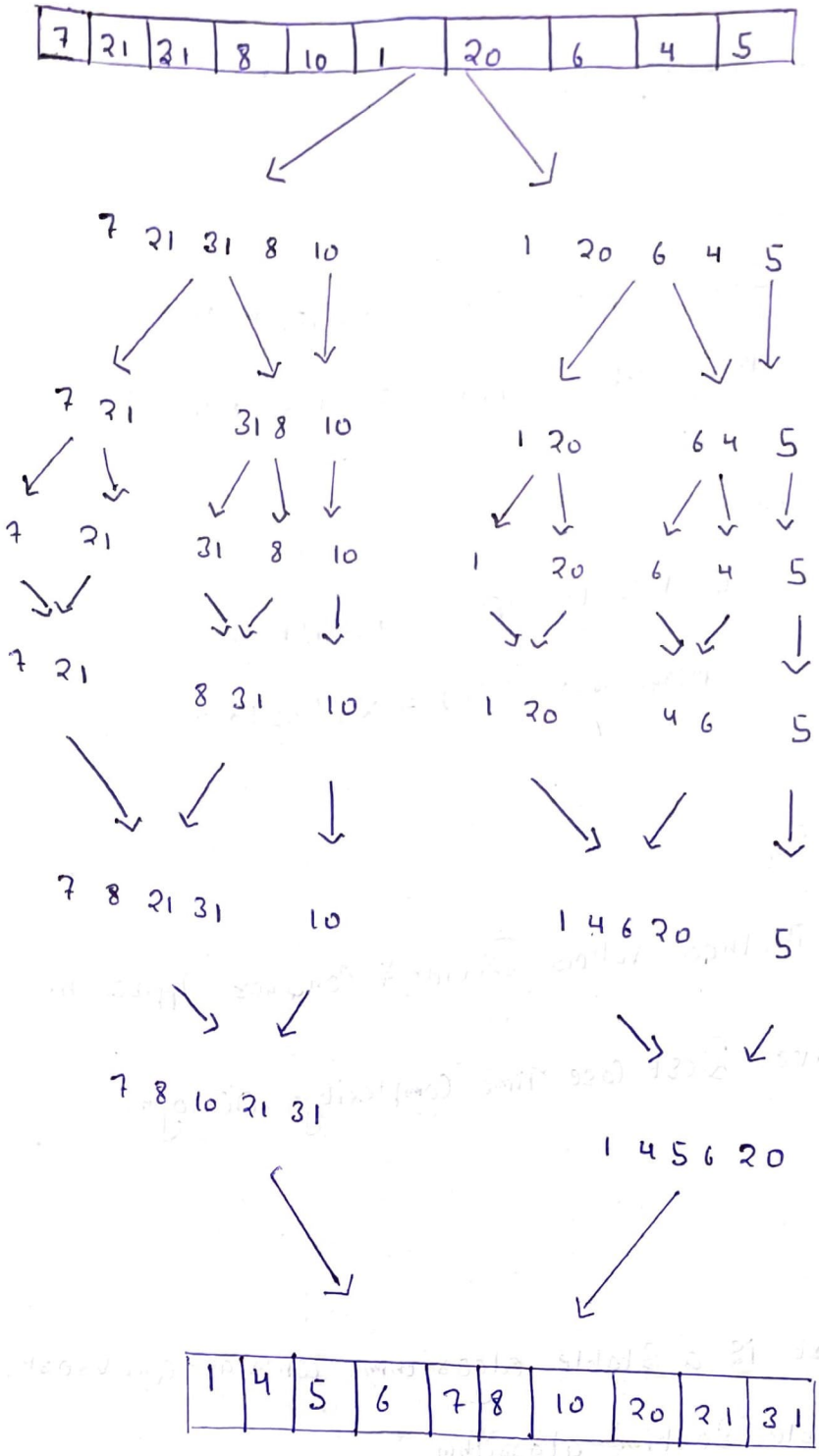
```
    if (flag == false)
    {
        cout << " No Such Pair Exist";
    }
    return 0;
}
```

Ans 8 = QuickSort is the fastest general-purpose sort.
In most practical situations, quicksort is the method of choice.
If stability is important and space is available, mergesort might be best.

Ans 9 = Inversion Count for an array indicates how far (or close) the array is from being sorted. If the array is already sorted, then the inversion count is 0, but if the array is sorted in the reverse order, the inversion count is the maximum.

Array arr[] = {2, 21, 31, 8, 10, 1, 20, 6, 4, 5}

For given array Total No. of Inversions = 31



Ans 10 - The best case for QuickSort will be when the middle element is picked as a pivot.

The worst case for Quick Sort is when array is sorted in either increasing or decreasing order.

Ans :- Recurrence Relation

Best Case

Quick Sort = $T(n) = 2T(n/2) + n$

Merge Sort = $T(n) = 2T(n/2) + n$

Worst Case

Quick Sort = $T(n) = T(n-1) + n$

Merge Sort = $T(n) = 2T(n/2) + n$

Similarities.

1. Both the Method follow Divide & Conquer Approach.
2. Both have Best Case Time Complexity $O(n \log n)$

Difference

1. Merge Sort is a Stable algorithm whereas QuickSort is not Stable Sorting algorithm
2. Worst Case T.C of Quick Sort is $O(n^2)$ whereas merge Sort have T.C $O(n \log n)$
3. The Quick Sort is internal Sorting method where the data is Sorted in main memory. Whereas merge Sort is external Sorting method.

Ans 12:-

```
void SelectionSort (int arr[], int n)   // Stable Version
{
    for (int i=0; i<n-1; i++)
    {
        int min = i;
        for (int j= i+1; j<n; j++)
        {
            if (arr[min] > arr[j])
            {
                min = j;
            }
        }
        int key = arr[min];
        while (min > i)
        {
            arr[min] = arr[min-1];
            min--;
        }
        arr[i] = key;
    }
}
```

Ans 13=

```
void bubble Sort (int arr[], int n)
{
    int i, j;
    bool Swapped;
    for (i=0; i<n-1; i++)
    {
        Swapped = false;
        for (j=0; j<n-i-1; j++)
        { if (arr[j] > arr[j+1])
          {Swap ( &arr[j], &arr[j+1]);
          Swapped = true;
        }
        if (Swapped == false)
        {
          break;
        }
    }
}
```

Ans 14= For this purpose we will use External Sorting technique, ex:- Merge Sort.

In Internal Sorting all the data is stored in main memory all the time while sorting.

In External Sorting data is stored in the slower external memory (usually a Hard Drive). In the sorting phase, chunks of data small enough to fit in main memory are read, sorted and written out to a temporary file.