Assignment No: 3

1. Searching Algorithms

Linear Search :

```c
#include <stdio.h>
int linearSearch(int a[], int n, int val) {
  // Going through array sequencially
  for (int i = 0; i < n; i++)
    {
        if (a[i] == val)
        return i+1;
    }
  return -1;
}
int main() {
  int a[] = {70, 40, 30, 11, 78, 41, 25, 14, 52}; // given array
  int val = 1; // value to be searched
  int n = sizeof(a) / sizeof(a[0]); // size of array
  int res = linearSearch(a, n, val); // Store result
  printf("The elements of the array are - ");
  for (int i = 0; i < n; i++)
  printf("%d ", a[i]);
  printf("\nElement to be searched is - %d", val);
  if (res == -1)
  printf("\nElement is not present in the array");
  else
  printf("\nElement is present at %d position of array", res);
  return 0;
}
```

Output:

```
PS C:\Users\vaishnavi> cd desktop
PS C:\Users\vaishnavi\desktop> gcc linear.c
PS C:\Users\vaishnavi\desktop> a
The elements of the array are - 70 40 30 11 78 41 25 14 52
Element to be searched is - 1
Element is not present in the array
PS C:\Users\vaishnavi\desktop> gcc linear.c
PS C:\Users\vaishnavi\desktop> a
The elements of the array are - 70 40 30 11 78 41 25 14 52
Element to be searched is - 41
Element is present at 6 position of array
PS C:\Users\vaishnavi\desktop>
```

Binary Search

```c
#include <stdio.h>
int binarySearch(int a[], int beg, int end, int val)
{
    int mid;
    if(end >= beg)
    {        mid = (beg + end)/2;
/* if the item to be searched is present at middle */
        if(a[mid] == val)
        {
            return mid+1;
        }
            /* if the item to be searched is smaller than middle, then it can
only be in left subarray */
        else if(a[mid] < val)
        {
            return binarySearch(a, mid+1, end, val);
        }
            /* if the item to be searched is greater than middle, then it can
only be in right subarray */
        else
        {
            return binarySearch(a, beg, mid-1, val);
        }
    }
    return -1;
}
int main() {
  int a[] = {11, 14, 25, 30, 40, 41, 52, 57, 70}; // given array
  int val = 40; // value to be searched
  int n = sizeof(a) / sizeof(a[0]); // size of array
  int res = binarySearch(a, 0, n-1, val); // Store result
  printf("The elements of the array are - ");
  for (int i = 0; i < n; i++)
  printf("%d ", a[i]);
  printf("\nElement to be searched is - %d", val);
  if (res == -1)
  printf("\nElement is not present in the array");
  else
  printf("\nElement is present at %d position of array", res);
  return 0;
}
```

Output:

```
PS C:\Users\vaishnavi> cd desktop
PS C:\Users\vaishnavi\desktop> gcc binary.c
PS C:\Users\vaishnavi\desktop> a
The elements of the array are - 11 14 25 30 40 41 52 57 70
Element to be searched is - 40
Element is present at 5 position of array
PS C:\Users\vaishnavi\desktop>
```

Jump Search :

```c
#include<stdio.h>

#include<math.h>
int min(int a, int b){
    if(b>a)
    return a;
    else
    return b;
}
int jumpsearch(int arr[], int x, int n)
{
    // Finding block size to be jumped
    int step = sqrt(n);

    // Finding the block where element is
    // present (if it is present)
    int prev = 0;
    while (arr[min(step, n)-1] < x)
    {
        prev = step;
        step += sqrt(n);
        if (prev >= n)
            return -1;
    }

    // Doing a linear search for x in block
    // beginning with prev.
    while (arr[prev] < x)
    {
        prev++;

        // If we reached next block or end of
        // array, element is not present.
        if (prev == min(step, n))
            return -1;
    }
    // If element is found
    if (arr[prev] == x)
        return prev;

    return -1;
}
int main()
{
    int arr[] = { 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377,
610};
    int x = 55;
    int n = sizeof(arr)/sizeof(arr[0]);
```

```
    int index = jumpsearch(arr, x, n);
    if(index >= 0)
    printf("Number is at %d index",index);
    else
    printf("Number is not exist in the array");
    return 0;
}
```

Output:

```
PS C:\Users\vaishnavi> cd desktop
PS C:\Users\vaishnavi\desktop> gcc jump.c
PS C:\Users\vaishnavi\desktop> a
Number is at 10 index
PS C:\Users\vaishnavi\desktop>
```