Tutorial 04

01. Using an appropriate array example, explain how many searching algorithms can be performed.

There are two searching algorithms.

1. Binary search
2. Linear search

Linear search :- Linear search is defined as a sequential search algorithm that starts at one end and goes through each element of a list until the desired element is found, otherwise the search continues till the end of the data set.

Ex:

21	12	3	4	10
0	1	2	3	4

key value = 4

i = 0

arr[0] = 21 21 != 4

i = 1

arr[1] = 12 12 != 4

i = 2

arr[2] = 3 3 != 4

i = 3

arr[3] = 4 4 = 4

∴ The key value is belongs to the 3rd index.

Binary search :- Binary search is defined as a searching algorithm used in a sorted array by repeatedly dividing the search interval in half.

Ex:

2	5	7	10	15	20	25
0	1	2	3	4	5	6

key element = 20

Step 1

mid point = $\frac{0+6}{2} = 3$ 10 < 20

Step 2

15	20	25
4	5	6

mid point = $\frac{4+6}{2} = 5$

arr[5] = 20

∴ index = 5

02. Compare and contrast linear search and binary search.

03. Write a c program for binary search.

```
#include <stdio.h>
int main()
{
    int c, first, last, middle, n, search, arr[100];

    printf("Enter number of elements\n");
    scanf("%d", &n);
```

04. Write a c program for linear search.

```
int main()
{
    int arr[100], search, c, n;
    printf("Enter number of elements in array\n");
    scanf("%d", &n);

    printf("Enter %d integer(s)\n", n);
    for(c=0; c<n; c++)
    scanf("%d", &arr[c]);

    printf("Enter a number to search\n");
    scanf("%d", &search);

    for(c=0; c<n; c++)
    {
      if(arr[c]==search) /* If required element is found */
      {
        printf("%d is present at location %d.\n", search, c+1);
        break;
      }
}
```

SAD WEB

```
    if (c == n)
        printf("%d isn't present in the array.\n", search);

    return 0;
}
```