



Sri
SAI RAM
ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY

West Tambaram, Chennai - 44

SAIRAM
DIGITAL RESOURCES



CS8391

**DATA STRUCTURES
(COMMON TO CSE & IT)**



UNIT NO 5

SEARCHING, SORTING AND HASHING TECHNIQUES

5.1 SEARCHING-LINEAR SEARCH-BINARY SEARCH

COMPUTER SCIENCE & ENGINEERING



SEARCHING

Linear Search

Linear Search Algorithm is used to find an item in the list. It starts searching for the item from the beginning of the list and continues till the end of the list until the item is found.

Binary Search

Binary search in **C** language to find a If the element to search is present in the list, then we print its location. The program assumes that the input numbers are in ascending order. n element in a sorted array

LINEAR SEARCH

```
#include <stdio.h>
void main()
{ int num;
  int i, keynum, found = 0;
  printf("Enter the number of elements ");
  scanf("%d", &num);
  int array[num];
  printf("Enter the elements one by one \n");
  for (i = 0; i < num; i++)
  {
    scanf("%d", &array[i]);
  }

  printf("Enter the element to be searched ");
  scanf("%d", &keynum);
```

```
/* Linear search begins */
for (i = 0; i < num ; i++)
{
  if (keynum == array[i] )
  {
    found = 1;
    break;
  }
}
if (found == 1)
  printf("Element is present in the array at
  position %d",i+1);
else
  printf("Element is not present in the array\n");
}
```

SORTING – BUBBLE SORT

The idea of bubble sort is to repeatedly move the largest element to the highest index position of the array.

- Bubble sort focuses on successive adjacent pairs of elements in the array, compares them, and either swaps them or not. In either case, after such a step, the larger of the two elements will be in the higher index position.
- The focus then moves to the next higher position, and the process is repeated. When the focus reaches the end of the array, the largest element will have "bubbled" from whatever its original position to the highest index position in the array.

Step by Step Example

[3 5 4 9 2]	[3 5 4 9 2] -- compare 3 to 5
[3 5 4 9 2]	[3 4 5 9 2] -- compare and swap 5 and 4
[3 4 5 9 2]	[3 4 5 9 2] -- compare 5 to 9
[3 4 5 9 2]	[3 4 5 2 9] -- compare and swap 2 and 9
Second Pass	
[3 4 5 9 2]	[3 4 5 2 9] -- compare 3 to 4
[3 4 5 2 9]	[3 4 5 2 9] -- compare 4 to 5
[3 4 5 2 9]	[3 4 2 5 9] -- compare and swap 2 and 5
[3 4 2 5 9]	[3 4 2 5 9] -- compare 5 to 9
Third Pass	
[3 4 2 5 9]	[3 4 2 5 9] -- compare 3 to 4
[3 4 2 5 9]	[3 2 4 5 9] -- compare and swap 2 and 4
[3 2 4 5 9]	[3 2 4 5 9] -- compare 4 to 5
[3 2 4 5 9]	[3 2 4 5 9] -- compare 5 to 9
Fourth Pass	
[3 2 4 5 9]	[2 3 4 5 9] -- compare and swap 2 and 3
[2 3 4 5 9]	[2 3 4 5 9] -- compare 3 to 4
[2 3 4 5 9]	[2 3 4 5 9] -- compare 4 to 5
[2 3 4 5 9]	[2 3 4 5 9] -- compare 5 to 9
Fifth Pass	
[2 3 4 5 9]	[2 3 4 5 9] -- compare 2 to 3
[2 3 4 5 9]	[2 3 4 5 9] -- compare 3 to 4
[2 3 4 5 9]	[2 3 4 5 9] -- compare 4 to 5
[2 3 4 5 9]	[2 3 4 5 9] -- compare 5 to 9

C Program To Sort data in ascending order using bubble sort

```
#include <stdio.h>
void main()
{
    int data[100],i,n,j,temp;
    printf("Enter the number of elements to be sorted: ");
    scanf("%d",&n);
    for(i=0;i<n;++i)
    {
        printf("%d. Enter element: ",i+1);
        scanf("%d",&data[i]);
    }
}
```

BUBBLE SORT – Contd..

```
for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j)
    {
        if(data[i]>data[j]) /* To sort in descending order, change > to < in this line. */
        {
            temp=data[i];
            data[i]=data[j];
            data[j]=temp;
        }
    }
}
printf("In ascending order: ");
for(i=0;i<n;++i)
    printf("%d ",data[i]);
}
```

BINARY SEARCH

```
#include <stdio.h>
void main()
{
int array[10];
int i, j, num, temp, keynum;
int low, mid, high;
printf("Enter the value of num
\n");
scanf("%d", &num);
printf("Enter the elements one
by one \n");
for (i = 0; i < num; i++)
{
scanf("%d", &array[i]);
}
printf("Input array elements
\n");
```

```
for (i = 0; i < num; i++)
{
printf("%d\n", array[i]);
}
/* Bubble sorting begins */
for (i = 0; i < num; i++)
{
for (j = 0; j < (num - i - 1); j++)
{
if (array[j] > array[j + 1])
{
temp = array[j];
array[j] = array[j + 1];
array[j + 1] = temp;
}
}
}
```


Continue...

```
printf("Sorted array is...\n");
for (i = 0; i < num; i++)
{
    printf("%d\n", array[i]);
}
printf("Enter the element to be
searched \n");
scanf("%d", &keynum);
/* Binary searching begins */
low = 1;
high = num;
do
{
    mid = (low + high) / 2;
    if (keynum < array[mid])
        high = mid - 1;
    else if (keynum > array[mid])
```

```
        low = mid + 1;
    } while (keynum != array[mid] && low
    <= high);
    if (keynum == array[mid])
    {
        printf("SEARCH SUCCESSFUL \n");
    }
    else
    {
        printf("SEARCH FAILED \n");
    }
}
```

VIDEO LINK

<https://www.youtube.com/watch?v=AqjVd6FVFbE&feature=youtu.be>

Sairam