

(Lecture
8.6)SearchingIt has following two
Types-

- 1) Linear Search
(Hit & Trial)
- 2) Binary Search
(Divide & Conquer)

Course Outline

- Searching
- Sorting
- Stacks
- Trees
- Graphs
- Asymptotic
- Notations

softbooks.m

Linear Search

- Complexity
- Algorithm & Design
- Algos
- Advance Topic & Linked List
- Singly
- Doubly
- Circular
- Queues
- Simple Queues
- Circular Queues

- Linear search has no exact mechanism or particular way to adopt

- We use hit & Trial method for Linear Search.

- Linear Search is useful for only small number of elements or system.

- In Linear Search, we have unsorted inputs.

Example Program for Linear Search

→ Program to find either number is present in array or not?

Code:-

```
#include <iostream>
using namespace std;
int main ( )
{
    int lgl, nmb;
    int arr [lgl];
    cout << "How many numbers you want to enter";
    cin >> lgl;

    for (int i=0; i<lgl; i++)
    {
        cout << "Enter number";
        cin >> arr [i];
    }

    cout << "Enter number you want to check";
    cin >> nmb;

    int check = 0;
```

softbooks.ml


```
for (int i = 0; i < arr.length; i++)
```

```
{  
    if (numbr == arr[i])
```

```
    {  
        check = 1;
```

```
        break;
```

```
    }
```

```
}
```

```
if (check == 1)
```

```
    cout << "Number found";
```

```
else
```

```
    cout << "Number not exist";
```

```
return 0;
```

```
}
```

Case for Linear Search

i) Best case:-

Number found in 1st. So $n = 1$

$\therefore (n = \text{number of elements})$

ii) Average case:-

Number found in mid so $(n/2)$

iii) Worst case:-

Number found at last or not found.
So (n)

softbooks.ml