

## 28 Binary search

- we use most efficient techniques to search the elements from an array
- It is known as binary search. Its main characteristics are that only works on sorted array list so it becomes easy to find and any information very fast
- It is also used to find the location of a given element or record in a list. If there is requirement to other related information then it is also possible.
- To find the location of a file in the computer directory we can use this searching techniques. If we are on the internet then it is not an easy task using linear search to find out the information about some records or files
- In binary search three things are performed low, high and middle. In which low represents lower limit of the list which is always 0, high represents upper limit of the list which is always and middle which is the average of low and high of the lower and upper limits. It is calculated by the following formula:

$$\text{mid} = (\text{low} + \text{high}) / 2$$

- we compare the middle element with key to be searched. If the value of middle element is greater than key, then key will exist in lower half of the list, take  $\text{high} = \text{mid} - 1$  and find new value of mid

$$\text{mid} = (\text{low} + \text{high}) / 2$$

- If the value of the middle element is less than key it means that now key value is to be searched on the lower side so now low is initialized with



$low = mid + 1$

- again middle value is searched  $mid = (low + high) / 2$ .

### \* ALGORITHM

binary\_search(element, n, arr)

where element  $\rightarrow$  represents the value we have to search in the list.

n  $\rightarrow$  represents number of elements in the list

arr  $\rightarrow$  represents list of the elements

Step-1 : [Initialize]

$low \leftarrow 0, high \leftarrow n-1, flag \leftarrow 1$

Step-2: Repeat Stepph step-4 while( $low \leq high$ )

Step-3 :  $mid \leftarrow (low + high) / 2$

Step-4 : if (element < arr[mid]) then

$high \leftarrow mid - 1$

else if (element > arr[mid]) then

$low \leftarrow mid + 1$

else if (element = arr[mid])

output "search is successful at" mid+1, "location"

flag = 0

return

Step-5: if (flag) then

output "search is unsuccessful"



## 3 binary search

```
int low = 0, high = n - 1, flag = 1, mid;
```

```
while (low <= high) {
```

```
    mid = (low + high) / 2;
```

```
    if (ch <= high) {
```

```
        if (ch <= arr[mid]) {
```

```
            high = mid - 1;
```

```
        }
```

```
    else if (arr[mid] < ch) {
```

```
        low = mid + 1;
```

```
    }
```

```
    else {
```

```
        printf("\n Element found at position %d  
        and element is %d\n", mid, ch);
```

```
        flag = 0;
```

```
        break;
```

```
    }
```

```
}
```

```
if (flag == 1) {
```

```
    printf("searched element not found");
```