Linear Search

Linear Search is a sequential searching algorithm where we start from one end and check every element of the list until the desired element is found.

* How Linear Search works?

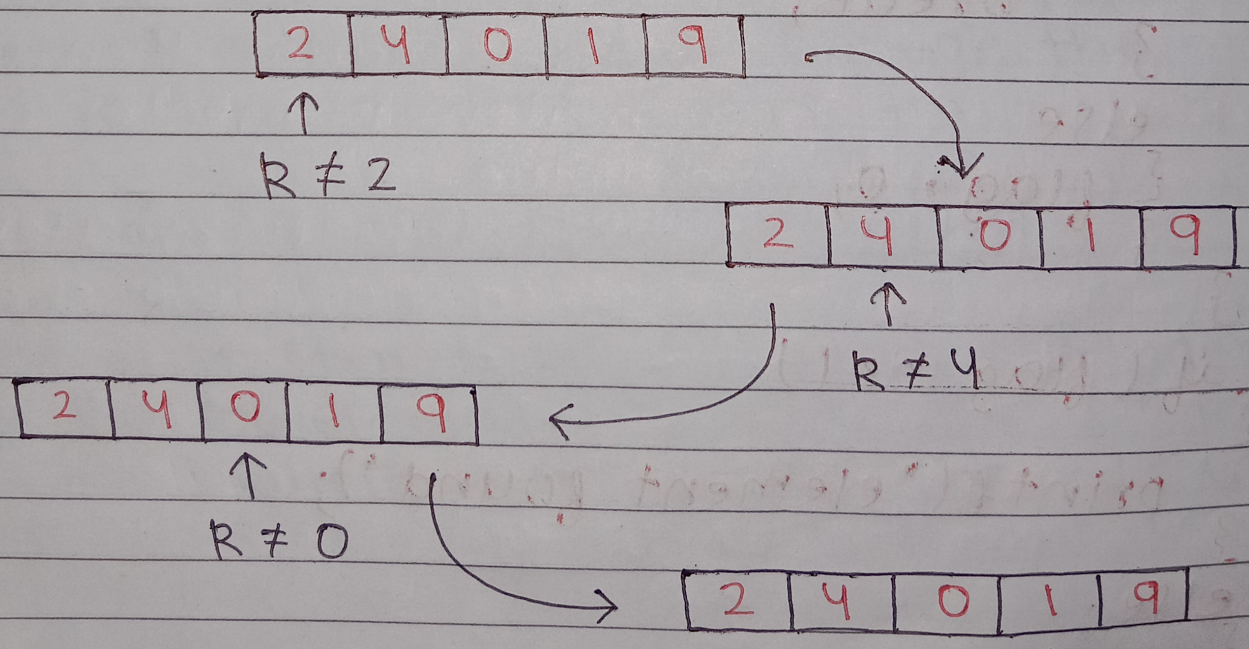
Let suppose, we have an array :-

2	4	0	1	9

} this is the array to be searched for.

Search item (R) = 1

Step 1, Start from the first element, compare R with each element of the array.



when "x == R", return the index, else print ("not found")

* Linear search :-

```
#include <stdio.h>
int main()
{
   int arr[] = {75, 58, 12, 19, 69, 71, 37,
                44, 99, 105 };
   int i, search, flag = 0;
   printf("enter element to search: ");
   scanf("%d", &search);

  for (i=0; i<10; i++)
  {
    if (arr[i] == search)
    {
        flag += 1;
        break;
    }
    else
    { flag = 0;
    }
  }
 if (flag == 1)
{
   printf("element found");
}
else
{
 printf("not found");
}
return 0;
}
```

Binary Search

Binary search is a searching algorithm for finding an element's position in a sorted array.

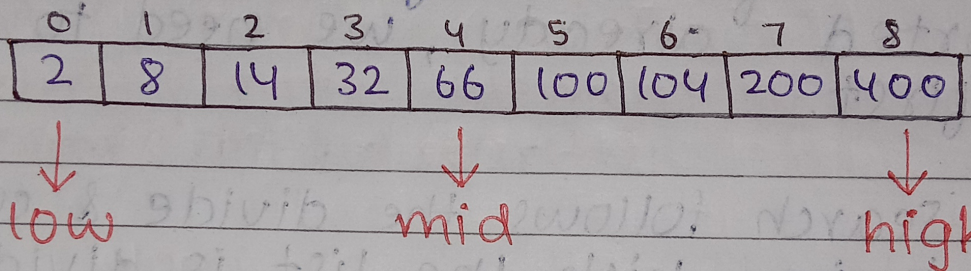
Binary search can be implement only on a sorted list of items. If the elements are not sorted already, we need to sort them first.

Binary search follows the divide & conquer approach in which the list is divided into two halves & the item is compared with the middle element of the list. If the match is found then, the location of the middle element is returned. Otherwise, we search into either of the halves depending upon the result produce through the match.

0	1	2	3	4	5	6	7	8
2	8	14	32	66	100	104	200	400

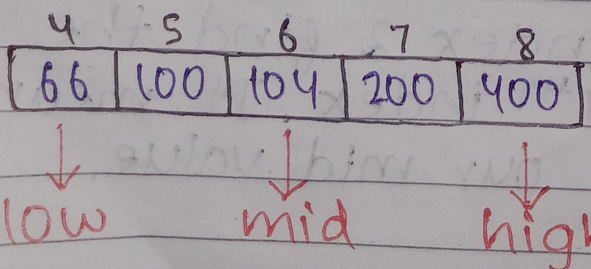
The first step is to make the 0th index as the lowest index & make the 8th index as the highest index & find the mid value, i.e., 0 + 8 / 2 = 4th index. It means, 4th index, i.e., 66 is our mid value.

Now, we compare our search element with the mid value, if the value match then we return the index otherwise we search left part or right part accor -ding to our search value & the mid value.

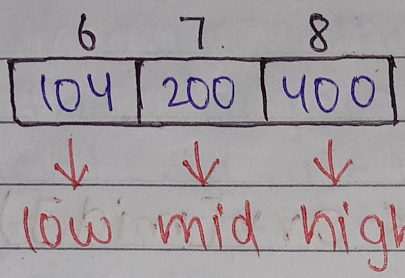


Search item : 200

Now, we check if mid value is eaual to search item, then we return the index of mid value, if its not eaual then we check that the search item is greater or lesser with the mid value.
In our case, the search item is greater than the mid value so now we check the right side array, i.e.,



Here also our search item is greater than the mid value, so we again check the right side array.



Now when we compare our search item with the mid value, they are equal & we return the index of our mid value, i.e., 7th index position.

* Binary Search :-

```
#include <stdio.h>
int main()
{
    int arr[] = {25, 27, 36, 41, 48, 55, 61, 67};
    int n = 8;
    int i, search, flag = 0, low, mid,
                                    high;

    low = 0;
    high = n-1;
    printf("enter item to search = ");
    scanf("%d", &search);

    while (low <= high)
{
    mid = (low + high)/2;
```

```
        if(search == arr[mid])
        {
            flag = 1;
            break;
        }
        else
        {
            if(search < arr[mid])
            {
                high = mid - 1;
            }
            else
            {
                low = mid + 1;
            }
        }
    }

    if("flag == 1")
    {
        printf("element found");
    }
    else
    {
        printf("not found");
    }
    printf("\n");
    return 0;
}
```