

Binary Search

Problem Description: Given a sorted integer array (in ascending order) and an element x . You need to search this element x in the given input array using binary search. Return the index of element in the input. If element is not found in the array, return -1.

Sample Input:

```
8
1 3 5 6 8 10 14 17
6
```

Sample Output:

```
3
```

How to approach?

Since the array is sorted (in increasing order), we can directly go to the middle element, compare it with the element we wish to find (x in this case) and then do the following.

1. If middle element is equal to x , return that element's index.
2. If middle element is less than x , we'll do the search in the right half of the array.
3. If middle element is more than x , we'll do the search in the left half of the array.

To implement this approach, we'll maintain the two pointers; start and end. Initially start would point to the first element of array, and end would point to the last element of the array.

We'll find the middle index by doing $(start+end)/2$ and assigning it to an element called `mid`. Then, we'll begin the search. First we'll compare `arr[mid]` with x . Three scenarios will arise:

1. `arr[mid] = x` -> This means we've found the element and we can return `mid` straight away.
2. `arr[mid] < x` -> This means that the element will not exist in the left half of the array, and may exist in the right half so we'll update `start` to `mid+1` and continue the search
3. `arr[mid] > x` -> This means that the element will not exist in the right half of the array, and may exist in the left half so we'll update `end` to `mid-1` and continue the search

The pseudo-code for this approach is shown on the next page.

```
function binarySearch(arr, x):
```

```
  start <- 0
```

```
  end <- arr.size - 1
```

```
  while(start <= end):
```

```
    mid <- (start + end)/2
```

```
    if (arr[mid] = x):
```

```
      return mid
```

```
    else if (arr[mid] < x):
```

```
      start <- mid + 1
```

```
    else:
```

```
      end <- mid - 1
```

```
  return -1
```