

Homework Questions of 17th february 2023

Q1 Apply binary search for elements in descending order

i/p \rightarrow 13 10 9 6 2 1

key = 6, key = 14

o/p \rightarrow found, not found

Algorithm

s = 0

e = size - 1

mid = s + (e - s) / 2

Run a loop until $s \leq e$

13	10	9	6	2	1
0	1	2	3	4	5
$mid = \frac{0+5}{2} = 2$					

arr[mid] = 9, key = 6

Cond \rightarrow arr[mid] > key, search in right i.e

$s = mid + 1;$

$$s = 2 + 1 = 3$$

13	10	9	6	2	1
0	1	2	3	4	5

$$s = 3$$

$$e = 5$$

$$\text{mid} = \frac{s+e}{2} = \frac{3+5}{2} = 4$$

$$\text{arr}[\text{mid}] = 2, \text{key} = 6$$

Cond2 \rightarrow $\text{arr}[\text{mid}] < \text{key}$, then search in left i.e. $e = \text{mid} - 1$

13	10	9	6	2	1
0	1	2	3	4	5

$$s = 3$$

$$e = 3$$

$$\text{mid} = \frac{s+e}{2} = 3$$

$$\text{arr}[\text{mid}] = 6, \text{key} = 6$$

Cond3 \rightarrow $\text{arr}[\text{mid}] == \text{key}$, hence return mid index.

Code

```
int binarySearch (vector<int> arr, int key){
```

```
    int s = 0;
```

```
    int e = arr.size() - 1;
```

```
    int mid = s + (e-s)/2;
```

```
    while (s <= e) {
```

```

if (arr[mid] == key) {
    return mid; //Return mid if
} element found.
else if (arr[mid] > key) {
    s = mid + 1; //Search in right
}
else {
    e = mid - 1; //Search in left
}
mid = s + (e-s)/2;
}
return -1; //If element not found
}

```

Q2 Find missing element in the array with the help of binary search.

i/p → 1 2 3 4 6
o/p → 5

Algorithm

1	2	3	4	6
0	1	2	3	4

First of all we have to care of extreme cases like

2	3	4
---	---	---

- 1) At 0th index 1 is not coming & hence 1 is the missing number.

1	2	3	, size = 4
---	---	---	------------

2) At $(\text{size}-1)$ th index i.e 3rd index 4
Should come but it is not there & hence
4 is the missing number.

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 6 \\ \{ & \{ & \{ & \{ & \} \end{array}$$

$$2-1=1 \quad 3-2=1 \quad 4-3=1 \quad 6-4=2$$

Here the difference
is 2 & hence
something is
wrong in this case.

$$s = 0$$

$$e = \text{size} - 1$$

$$\text{mid} = s + (e-s)/2$$

We have to run the loop until $e-s > 1$.

$$\checkmark \quad s = 0$$

$$e = 4$$

$$\text{mid} = \frac{0+4}{2} = 2$$

$$\text{arr}[\text{mid}] = 3$$

$$\text{arr}[s] - s = 1 - 0 = 1$$

$$\text{arr}[\text{mid}] - \text{mid} = 3 - 2 = 1$$

$$(\text{arr}[s] - s) \neq \text{arr}[\text{mid}] - \text{mid} \Rightarrow \text{False}$$

If it was true then this means the
missing element was on the left & hence
search in left by $e=\text{mid}$.

$$\text{arr}[e] - e = 6 - 4 = 2$$

$$\text{arr}[mid] - mid = 3 - 2 = 1$$

$(\text{arr}[e] - e) \neq \text{arr}[mid] - mid \Rightarrow$ Hence this is true & this means the missing element is on the right & hence search in the right.

$s = mid ;$ // To search in right

$$\text{if } s = 2$$

$$e = 4$$

$$mid = \frac{2+4}{2} = 3$$

$$\text{arr}[mid] = 4$$

$$\text{arr}[s] - s = 3 - 2 = 1$$

$$\text{arr}[mid] - mid = 4 - 3 = 1$$

1st condition is false ($1 \neq 1$)

$$\text{arr}[e] - e = 6 - 4 = 2$$

The 2nd condition is true & hence again we have to search in right.

$s = mid ;$

$$\text{if } s = 3$$

$$e = 4$$

Here $e - s = 1$ which is not greater than 1.

Hence we come out of the loop & return

$$\text{arr}[s] + 1$$

$$\text{arr}[3] + 1 \Rightarrow 4 + 1 = 5 \text{ Ans}$$

Code

```
int findElement (vector <int> arr) {  
    int s = 0 ;  
    int e = arr.size () - 1 ;  
    int mid = s + (e-s)/2 ;  
  
    while ((e-s)>1) {  
        if (arr [s] - s != arr [mid]-mid){  
            e=mid ;  
        }  
        else if (arr [e] - e != arr [mid]-mid){  
            s=mid ;  
        }  
        mid = s + (e-s)/2 ;  
    }  
    return arr [s] + 1 ;  
}
```