"श्री कृष्ण गोविंद हरे मुरारी, हे नाथ नारायण वासुदेवा"

Data Structure and Algorithm

By

Malay Tripathi

BINARY TREE

SEARCH ELEMENT IN ROTATED SORTED ARRAY - II. [DUPLICATES].

→ arr[] = { 7 8 1 2 3 3 3 4 5 6 } target = 3.

in this, you have to find target exists or not. → you don't need to return index.

Step 1: Identify the sorted part

```
          0 1 2 3 4 5 6 7
arr[] = [ 8 7 1 2 3 4 4 5 ]
          ↑   ↑       ↑
         low  mid    high
```

→ Right part is Sorted Because element at 3 index < element at 7 index.

→ Left part is not sorted Because element at 0 index > element at 2 index or mid index.

```
arr[] = [ 3 1, 2, 3, 3, 3, 3 ]
          ↑    ↑        ↑
         low  mid      high
```

here it becomes difficult to find out which part is sorted. because R-H-S is equal mid = high (3 = 3) and LHS value at low index. is 3 which is equal to value at mid index that is 3, so we cant say whether the left hand side is sorted or not.

we cant say anything whether, value is sorted or not because value at high, mid and low all are 3.

DSA BY Malay Tripathi

Scanned with OKEN Scanner

∴ if target=1.

try to trim down the condition :→ which is

arr[low] = arr[mid] = arr[high].

⇒ always compare what is at the mid to the target.
So, we can say that, if arr[mid] is not equal to the target, neither array[low] nor array[high] will be equal to target. so we can shrink our search space.

```
if (arr[low] == arr[mid] && arr[mid] == arr[high])
        low = low+1;
        high = high-1;
        continue;
    }
```

C++ code.

```
bool searchInRotate(vector<int>& arr, int k) {
  int n=arr.size();
  int low=0, high=n-1;
  while(low<=high){
    int mid =(low+high)/2;
    if(arr[mid]==k) return true;
    if(arr[low]==arr[mid] && arr[mid]==arr[high]) {
        low++, high--;
        continue;
    }
    if(arr[low]<=arr[mid]) {
      if(arr[low]<=k && k<=arr[mid]){
        high=mid-1;
      }else{
        low=mid+1;
      }
    }else{
      if(arr[mid]<=k && k<=arr[high]){
        low=mid+1;
      }else{
        high=mid-1;
      }
    }
  }
  return false;
}
```

DSA by Malay Tripathi

Scanned with OKEN Scanner