7] +

Find target in mountain Array

arr = [0 1, 1 2, 2 3, 3 4, 4 5, 5 3, 6 1]
target = 3 peak

- Find Peak element => index 4

- Binary search in ascending part of array by 0 to peak (0, 4)
  search (arr, target, 0, peak)

- if not found, than apply binary search in descending part of array (5, 6)
  search (arr, target, peak+1, length)

* = Let's code

FindPick => we can use code of Q.5 (previous)

```
search(int[] arr, int target) {
  int peak = peakInMountain(arr);

  // First try in ascenting part
  int firstTry = orderAgnasticBS(arr, target
                 , 0, peak, true);
  if (firstTry != -1) {
    return firstTry; // we found target
  }                  in first try
```

// if not found in first try look in desc part
return orderOgnasticBs(arr, target, peak+1, arr.length-1, false);

}

=) orderOgnasticBs(int[] arr, int target, int start, int end, boolean isFirst){

while(start<=end){

int mid = start + (end-start)/2;
if(target == arr[mid]){ return mid;}
if(isfirst){ // look in ascending part

if(~~arr[end]~~ target > arr[mid]){
start = mid+1;
}else{
end = mid-1;
}
}else{ // look in descending part

if(target > arr[mid]){
end = mid-1;
}else{
start = mid+1;
}

}
}

return -1; //target not found
}