DAY 8

class Solution { public int findInMountainArray(int target, MountainArray mountainArr) { int n = mountainArr.length(); int start = 0; int end = n - 1;

// Step 1: Find the peak index  
 while (start < end) {  
 int mid = start + (end - start) / 2;  
 int mid\_val = mountainArr.get(mid);  
 int next\_val = mountainArr.get(mid + 1);  
  
 if (mid\_val < next\_val) {  
 start = mid + 1;  
 } else {  
 end = mid;  
 }  
 }  
  
 int peak = start;  
  
 // Step 2: Binary search in the increasing part  
 int result = binarySearch(mountainArr, 0, peak, target, true);  
 if (result != -1) {  
 return result;  
 }  
  
 // Step 3: Binary search in the decreasing part  
 return binarySearch(mountainArr, peak + 1, n - 1, target, false);  
}  
  
public int binarySearch(MountainArray arr, int start, int end, int target, boolean ascending) {  
 while (start <= end) {  
 int mid = start + (end - start) / 2;  
 int mid\_val = arr.get(mid);  
  
 if (mid\_val == target) {  
 return mid;  
 }  
  
 if (ascending) {  
 if (target < mid\_val) {  
 end = mid - 1;  
 } else {  
 start = mid + 1;  
 }  
 } else {  
 if (target > mid\_val) {  
 end = mid - 1;  
 } else {  
 start = mid + 1;  
 }  
 }  
 }  
  
 return -1;  
}

}