Mountain Array Leetcode:

class Solution {

    public int findInMountainArray(int target, MountainArray mountainArr) {

        int n = mountainArr.length();

        int start = 0, end = n - 1;

        while (start < end) {

            int mid = start + (end - start) / 2;

            if (mountainArr.get(mid) < mountainArr.get(mid + 1)) {

                start = mid + 1;

            } else {

                end = mid;

            }

        }

        int peak = start;

        int result = binarySearch(mountainArr, 0, peak, target, true);

        if (result != -1) {

            return result;

        }

        return binarySearch(mountainArr, peak + 1, n - 1, target, false);

    }

    private int binarySearch(MountainArray arr, int start, int end, int target, boolean ascending) {

        while (start <= end) {

            int mid = start + (end - start) / 2;

            int midVal = arr.get(mid);

            if (midVal == target) {

                return mid;

            }

            if (ascending) {

                if (target < midVal)

                {

                    end = mid - 1;

                }

                else

                {

                    start = mid + 1;

                }

            } else {

                if (target > midVal)

                {

                     end = mid - 1;

                }

                else

                {

                    start = mid + 1;

                }

            }

        }

        return -1;

    }

}