```
 1 function solve(A) {
 2     //Early returns
 3     if (A.length == 2)
 4         return 2;
 5
 6     let nodes = [];
 7
 8     debugger;
 9
10     for (let i = 0; i < A.length; i++) {
11
12         if (isPeak(i)) {
13             CheckInsert(i, 'P', A[i]);
14         } else {
15
16             if (isValley(i)) {
17                 CheckInsert(i, 'V', A[i]);
18             }
19         }
20     }
21     return ' FINAL: ' + findResults();
22
23 //-------------------------------------
24     function CheckInsert(i, type, val) {
25         if (i > 0) {
26             let prev = nodes[nodes.length - 1
   ];
27             if (prev.type === type &&
28                 prev.value === val) {
29                 nodes[nodes.length - 1].
   indexes.push(i);
30
31                 console.log('Resusing '+type)
32                 return;
33             }
```

```
34
35          }
36          console.log('Adding '+type)
37          nodes.push({
38              type: type,
39              indexes: [i],
40              value: A[i]
41          });
42          return;
43      }
44
45      function findResults() {
46          let maxDiff = 0;
47
48          let prevPeak;
49          let nextPeak;
50
51          for (let i = 0; i < nodes.length; i
    ++) {
52              if (nodes[i].type === 'P')
53                  continue;
54
55              if (i === 0)
56                  prevPeak = {type: 'P',
    indexes: [0]}
57              else
58                  prevPeak = nodes[i - 1];
59
60              if (i === nodes.length - 1) //at
    end
61                  nextPeak =  nodes[i]; //use
    current
62              else if (i < nodes.length - 1)
63                  nextPeak = nodes[i + 1];
64              else prevPeak = {type: 'P',
```

```
64 indexes: [0]}
65
66             let diff = Math.abs(Math.max(...
   nextPeak.indexes) - Math.min(...prevPeak.
   indexes));
67
68             console.log(`diff ${diff} for
   node ${i}  `);
69             if (diff > maxDiff)
70                 maxDiff = diff
71         }
72         maxDiff = maxDiff + 1;
73         console.log('Max Diff ' + maxDiff);
74         return maxDiff;
75     }
76
77 //TODO When elements are SAME At the BEGIN/
   END??
78     function isPeak(i) {
79         let num = A[i];
80
81         //to Pass: GREATER than LEFT /RIGHT
   NEIGHBORS!
82         if ((i - 1) >= 0 && num < A[i - 1
   ] ) { //LEFT
83             return false;
84         }
85
86         if ((i + 1) < A.length && num < A[i
    + 1] ) { //RIGHT
87             return false;
88         }
89         return true;
90     }
91
```

```
 92      function isValley(i) {
 93          let num = A[i];
 94
 95          //to Pass: GREATER than LEFT /RIGHT
    NEIGHBORS!
 96          if ((i - 1) >= 0 && num > A[i - 1
    ] ) { //LEFT
 97              return false;
 98          }
 99
100          if ((i + 1) < A.length && num > A[i
     + 1] ) { //RIGHT
101              return false;
102          }
103          return true;
104      }
105
106 }
107
```