Editorial: UpDown Sequences (ZCO 2019)

Problem Link : https://www.codechef.com/ZCOPRAC/problems/UPDOWSEQ

Let us define two types of sequences :-
1) **Up-Down Sequence (UD)** : A sequence of the form a[i] <= a[i+1] >= a[i+2]....i.e. the sequence starts by increasing first.
2) **Down-Up Sequence (DU)** : A sequence of the form a[i] >= a[i+1] <= a[i+2]....i.e. the sequence starts by decreasing first.

Notice that it is always possible to insert the integer in the sequence regardless of the inequality sign we want to impose on it. Why ?
Suppose the UD sequence breaks at a[k]. That means we have a[k] < a[k+1] when we expected it to be a[k] >= a[k+1]. So, we can insert an element x = a[k]. Now, a[k] >= x <= a[k+1] satisfies.
Similarily, we can also insert an integer if sequence broke up at a[k] and a[k] > a[k+1] when we expected it to be a[k] <= a[k+1].
But we don't have to actually insert anything. We have to output just the length of the longest UD sequence possible.

Now, Let us define **f (i , 1)** to be the longest UD sequence beginning at i and **f (i , 2)** to be the longest DU sequence beginning at i without inserting anyting.

$$\mathbf{f\,(i\,,\,state)} \;=\; \begin{cases} 1 \;;\; i = N \\ 1 \;;\; state = 1 \;\&\; a[i] > a[i+1] \\ 1 \;;\; state = 2 \;\&\; a[i] < a[i+1] \\ 1 + f\,(i+1\,,\,2) \;;\; state = 1 \;\&\; a[i] <= a[i+1] \\ 1 + f\,(i+1\,,\,1) \;;\; state = 2 \;\&\; a[i] >= a[i+1] \end{cases}$$

We will use Dynamic Programming to calculate the above result. Let the result be stored in **dp[ ][ ]**.

Now, Observe that a UD sequence is a combination of :-
1) UD_1 + x + UD_2 if the length of the UD_1 is odd
2) UD_1+ x + DU_1 if the length of the UD_1 is even
where x is the inserted element.

So for each i, we calculate the length of the longest UD sequence starting at i. Let the length be x.
If x is odd, we insert an element there (theoretically) and calculate the length of longest UD sequence starting at i+x. The longest UD sequence beginning at i after inserting an element is therefore **dp[i][1] + 1 + dp[i+x][1]**.
If x is even, we insert an element there (theoretically) and calculate the length of longest DU sequence starting at i+x. The longest UD sequence beginning at i after inserting an element is therefore **dp[i][1] + 1 + dp[i+x][2]**.
So the final answer is maximum among all i.

Implementation : https://pastebin.com/v57uceH9