Aesthetic Tree Problem

Problem Statement:
A few years ago, Bob planted N trees in a row in his backyard. Over the years, the trees have grown at different rates. The tree heights are stored in an array T. As a perfectionist, Bob wants his backyard to be aesthetic and that only occurs when the trees alternate in height. For example, $T = [1, 2, 1, 4, 3]$ would be considered aesthetic while $T = [1, 2, 3]$ would not be. Note that adjacent trees also cannot be of the same height so $T = [2, 2]$ would not be aesthetic either. Since Bob is lazy, he wants the find the minimum number of **cuts** needed to make his backyard aesthetic. Give an $O(n)$ algorithm to solve this problem.

Solution:
A valley occurs when a tree is shorter than both of its neighbors. If N is even, we desire $N/2$ valleys and if N is odd, we desire either $(N + 1)/2$ even valleys (starting at the first tree) or $(N - 1)/2$ odd valleys (starting at the second tree). We can simply loop through the array and count the number of even and odd valleys that occur (even valleys occur on the even indices and vice versa). If N is odd, we will subtract 0.5 from the number of even valleys and add 0.5 to the number of odd valleys. Finally, we will take the maximum of the number of even and odd valleys and subtract that from $N/2$ to get our answer.