Geek Jump

Geek wants to climb from the 0th stair to the (n-1)th stair. At a time the Geek can climb either one or two steps. A height[N] array is also given. Whenever the geek jumps from stair i to stair j, the energy consumed in the jump is abs(height[i]-height[j]), where abs() means the absolute difference. return the minimum energy that can be used by the Geek to jump from stair 0 to stair N-1.

Example:
Input:
n = 4
height = {10 20 30 10}
Output:
20
Explanation:
Geek jump from 1st to 2nd stair(|20-10| = 10 energy lost).
Then a jump from the 2nd to the last stair(|10-20| = 10 energy lost).
so, total energy lost is 20 which is the minimum.

Your Task:
You don't need to read input or print anything. Your task is to complete the function **MinimumEnergy()** which takes the array **height**, and integer **n,** and returns the minimum energy that is lost.

Expected Time Complexity: O(n)
Expected Space Complexity: O(n)

Constraint:
1<=n<=100000
1<=height[i]<=1000

Solution:

```
class Solution{

    public static int dp1(int arr[],int n,int dp[]){

        if(n==0){

            return 0;

        }

        if(dp[n]!=-1){

            return dp[n];

        }
        int right = Integer.MAX_VALUE;
        int left = dp1(arr,n-1,dp)+Math.abs(arr[n]-arr[n-1]);
        if(n>1)
        right = dp1(arr,n-2,dp)+Math.abs(arr[n]-arr[n-2]);

        return dp[n] = Math.min(left,right);
    }

    public int minimumEnergy(int arr[],int N){
        //code here
        int l = arr.length;
        // int dp[] = new int[l];
        // Arrays.fill(dp,-1);
        // int res = dp1(arr,arr.length-1,dp);
        int prev= 0;
        int prev2 = 0;
        int i;
        int res=0;
        for(i=1;i<l;i++){
            int left = prev+Math.abs(arr[i]-arr[i-1]);
            int right=Integer.MAX_VALUE;
            if(i>1)
            right =  prev2+Math.abs(arr[i]-arr[i-2]);
            res = Math.min(left,right);
            prev2 = prev;
            prev = res;
        }
        // System.out.println(res);

        return res;
    }
}
```