1671. Minimum Number of Removals to Make Mountain Array

You may recall that an array `arr` is a **mountain array** if and only if:

- `arr.length >= 3`
- There exists some index `i` (**0-indexed**) with `0 < i < arr.length - 1` such that:
    - `arr[0] < arr[1] < ... < arr[i - 1] < arr[i]`
    - `arr[i] > arr[i + 1] > ... > arr[arr.length - 1]`

Given an integer array `nums`, return *the* ***minimum*** *number of elements to remove to make* `nums` *a* ***mountain array***.

Example 1:

```
Input: nums = [1,3,1]
Output: 0
Explanation: The array itself is a mountain array so we do not need to
remove any elements.
```

Example 2:

```
Input: nums = [2,1,1,5,6,2,3,1]
Output: 3
Explanation: One solution is to remove the elements at indices 0, 1, and
5, making the array nums = [1,5,6,3,1].
```

Example 3:

```
Input: nums = [4,3,2,1,1,2,3,1]
Output: 4
```

Example 4:

```
Input: nums = [1,2,3,4,4,3,2,1]
Output: 1
```

```
def minimumMountainRemovals(self, arr: List[int]) -> int:
        lis = [1]*len(arr)
        lis[0] = 1
        lds = [1]*len(arr)
        # lds[-1]=1
        # lis.append(1)
        for i in range(1,len(arr)):
            temp = 0
            for j in range(i):
                if arr[j]<arr[i]:
                    temp = max(temp,lis[j])
            lis[i] = temp+1

        for i in range(len(arr)-1,-1,-1):
            temp = 0
            for j in range(len(arr)-1,i,-1):
                if arr[j]<arr[i]:
                    temp = max(temp,lds[j])
            lds[i] = temp+1
        maxL = 0
        # print(lis)
        # print(lds)
        for i in range(len(lis)):
            if lis[i]+lds[i]-1>maxL and lis[i]>=2 and lds[i]>=2:
                maxL = lis[i]+lds[i]-1
        return len(arr)-maxL
```