Problem 1246: Palindrome Removal

Problem Information

Difficulty: Hard
Acceptance Rate: 46.25%
Paid Only: Yes
Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `arr`.

In one move, you can select a **palindromic** subarray `arr[i], arr[i + 1], ..., arr[j]` where `i <= j`, and remove that subarray from the given array. Note that after removing a subarray, the elements on the left and on the right of that subarray move to fill the gap left by the removal.

Return _the minimum number of moves needed to remove all numbers from the array_.

Example 1:

Input: arr = [1,2] **Output:** 2

Example 2:

Input: arr = [1,3,4,1,5] **Output:** 3 **Explanation:** Remove [4] then remove [1,3,1] then remove [5].

Constraints:

* `1 <= arr.length <= 100` * `1 <= arr[i] <= 20`

Code Snippets

C++:

```
class Solution {
public:
int minimumMoves(vector<int>& arr) {

}
};
```

Java:

```
class Solution {
public int minimumMoves(int[] arr) {

}
}
```

Python3:

```
class Solution:
def minimumMoves(self, arr: List[int]) -> int:
```