Description | Solution | Discuss (60) | Submissions

Java | Autocomplete

1246. Palindrome Removal

Hard 247 9 Add to List Share

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`

Accepted 8,103 | Submissions 17,677

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Microsoft | 6

Related Topics

Array Dynamic Programming

Hide Hint 1

Use dynamic programming.

Hide Hint 2

Let dp[i][j] be the solution for the sub-array from index i to index j.

Hide Hint 3

Notice that if we have S[i] == S[j] one transition could be just dp(i + 1, j + 1) because in the last turn we would have a palindrome and we can extend this palindrome from both sides, the other transitions are not too difficult to deduce.

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

    }
}
```