Our Solution(s) Run Code

Solution 1 Solution 2

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.Arrays;

class Program {
  // O(n^3) time | O(n^2) space
  public static int palindromePartitioningMinCuts(String str) {
    boolean[][] palindromes = new boolean[str.length()][str.length()];
    for (int i = 0; i < str.length(); i++) {
      for (int j = i; j < str.length(); j++) {
        palindromes[i][j] = isPalindrome(str.substring(i, j + 1));
      }
    }
    int[] cuts = new int[str.length()];
    Arrays.fill(cuts, Integer.MAX_VALUE);
    for (int i = 0; i < str.length(); i++) {
      if (palindromes[0][i]) {
        cuts[i] = 0;
      } else {
        cuts[i] = cuts[i - 1] + 1;
        for (int j = 1; j < i; j++) {
          if (palindromes[j][i] && cuts[j - 1] + 1 < cuts[i]) {
            cuts[i] = cuts[j - 1] + 1;
          }
        }
      }
    }
    return cuts[str.length() - 1];
  }

  public static boolean isPalindrome(String str) {
    int leftIdx = 0;
    int rightIdx = str.length() - 1;
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
class Program {
  public static int palindromePartitioningMinCuts(String str) {
    // Write your code here.
    return -1;
  }
}
```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.