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

using System;

public class Program {
  // O(n^3) time | O(n^2) space
  public static int PalindromePartitioningMinCuts(string str) {
    bool[,] palindromes = new bool[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 - i));
      }
    }
    int[] cuts = new int[str.Length];
    Array.Fill(cuts, Int32.MaxValue);
    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 bool IsPalindrome(string str) {
    int leftIdx = 0;
    int rightIdx = str.Length - 1;
    while (leftIdx < rightIdx) {
      if (str[leftIdx] != str[rightIdx]) {
        return false;
      }
      leftIdx++;
      rightIdx--;
    }
    return true;
  }
}

```