

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <climits>
5 using namespace std;
6
7 bool isPalindrome(string s);
8
9 // O(n^3) time | O(n^2) space
10 int palindromePartitioningMinCuts(string s) {
11     vector<vector<bool>> palindromes(s.length(), vector<bool>(s.length(), false));
12     for (int i = 0; i < s.length(); i++) {
13         for (int j = i; j < s.length(); j++) {
14             palindromes[i][j] = isPalindrome(s.substr(i, j + 1 - i));
15         }
16     }
17     vector<int> cuts(s.length(), INT_MAX);
18     for (int i = 0; i < s.length(); i++) {
19         if (palindromes[0][i]) {
20             cuts[i] = 0;
21         } else {
22             cuts[i] = cuts[i - 1] + 1;
23             for (int j = 1; j < i; j++) {
24                 if (palindromes[j][i] && cuts[j - 1] + 1 < cuts[i]) {
25                     cuts[i] = cuts[j - 1] + 1;
26                 }
27             }
28         }
29     }
30     return cuts[s.length() - 1];
31 }
32
33 bool isPalindrome(string s) {
34     int leftIdx = 0;
35     int rightIdx = s.length() - 1;
36     while (leftIdx < rightIdx) {
37         if (s[leftIdx] != s[rightIdx]) {
38             return false;
39         }
40         leftIdx++;
41         rightIdx--;
42     }
43     return true;
44 }
45
```

