

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(n^3) time | O(n^2) space
4 function palindromePartitioningMinCuts(string) {
5   const palindromes = new Array(string.length).fill(1).map(row => []);
6   for (let i = 0; i < string.length; i++) {
7     for (let j = i; j < string.length; j++) {
8       palindromes[i][j] = isPalindrome(string.slice(i, j + 1));
9     }
10  }
11  const cuts = new Array(string.length);
12  cuts.fill(Infinity);
13  for (let i = 0; i < string.length; i++) {
14    if (palindromes[0][i]) {
15      cuts[i] = 0;
16    } else {
17      cuts[i] = cuts[i - 1] + 1;
18      for (let j = 1; j < i; j++) {
19        if (palindromes[j][i] && cuts[j - 1] + 1 < cuts[i]) {
20          cuts[i] = cuts[j - 1] + 1;
21        }
22      }
23    }
24  }
25  return cuts[cuts.length - 1];
26 }
27
28 function isPalindrome(string) {
29   let leftIdx = 0;
30   let rightIdx = string.length - 1;
31   while (leftIdx < rightIdx) {
32     if (string[leftIdx] !== string[rightIdx]) return false;
33     leftIdx++;
34     rightIdx--;
35   }
36   return true;
37 }
38
39 exports.palindromePartitioningMinCuts = palindromePartitioningMinCuts;
40
```

