

Solution 1

Solution 2

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

2

3 import java.util.Arrays;

4

5 class Program {

6 // O(n^2) time | O(n^2) space

7 public static int palindromePartitioningMinCuts(String str) {

8 boolean[][] palindromes = new boolean[str.length()][str.length()];

9 for (int i = 0; i < str.length(); i++) {

10 for (int j = 0; j < str.length(); j++) {

11 if (i == j) {

12 palindromes[i][j] = true;

13 } else {

14 palindromes[i][j] = false;

15 }

16 }

17 }

18 for (int length = 2; length < str.length() + 1; length++) {

19 for (int i = 0; i < str.length() - length + 1; i++) {

20 int j = i + length - 1;

21 if (length == 2) {

22 palindromes[i][j] = (str.charAt(i) == str.charAt(j));

23 } else {

24 palindromes[i][j] = (str.charAt(i) == str.charAt(j) && palindromes[i + 1][j - 1]);

25 }

26 }

27 }

28 int[] cuts = new int[str.length()];

29 Arrays.fill(cuts, Integer.MAX_VALUE);

30 for (int i = 0; i < str.length(); i++) {

31 if (palindromes[0][i]) {

32 cuts[i] = 0;

33 } else {

34 cuts[i] = cuts[i - 1] + 1;

35 for (int j = 1; j < i; j++) {

36 if (palindromes[j][i] && cuts[j - 1] + 1 < cuts[i]) {

37 cuts[i] = cuts[j - 1] + 1;

38 }

39 }

40 }

41 }

42 return cuts[str.length() - 1];

43 }

44 }

45

