Our Solution(s)

Run Code

Solution 1   Solution 2

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

package main

import "math"

// O(n^2) time | O(n^2) space
func PalindromePartitioningMinCuts(str string) int {
  palindromes := make([][]bool, len(str))
  for i := range palindromes {
    palindromes[i] = make([]bool, len(str))
  }
  for i := range str {
    palindromes[i][i] = true
  }
  for length := 2; length < len(str)+1; length++ {
    for i := 0; i < len(str)-length+1; i++ {
      j := i + length - 1
      if length == 2 {
        palindromes[i][j] = (str[i] == str[j])
      } else {
        palindromes[i][j] = (str[i] == str[j] && palindromes[i+1][j-1]
      }
    }
  }
  cuts := make([]int, len(str))
  for i := range cuts {
    cuts[i] = math.MaxInt32
  }
  for i := range str {
    if palindromes[0][i] {
      cuts[i] = 0
    } else {
```

Your Solutions

Run Code

Solution 1   Solution 2   Solution 3

```go
package main

func PalindromePartitioningMinCuts(str string) int {
  // Write your code here.
  return -1
}
```

Our Tests

Custom Output

Submit Code

Run or submit code when you're ready.