

# COMS W4995 Parallel Functional Programming

## Project Proposal: Palindrome Partitioning

Jesse Chong  
jlc2332@columbia.edu  
November 22 2021

### INTRODUCTION

Palindrome partitioning<sup>1</sup> is an optimization problem where we must split a string into palindrome substrings and then return the minimum number of “cuts” needed to complete the partition.

To clarify what “cuts” are, take this for example: the string “abracecar” would require 2 cuts to be minimally partitioned into the substrings “a”, “b”, and “racecar”. Also strings can only be made of lowercase alphabetical characters.

This problem is a variation of the matrix chain multiplication problem.<sup>2</sup>

### IMPLEMENTATION

The sequential solution to this problem would be to iterate over the given string for every possible substring and check that we found the longest possible palindromic substring. For example, in the “abracecar” example, partitioning on the substring “racecar” is the best choice while partitioning every character by itself in “racecar” (e.g., “r”, “a”, “c”.. etc.) would be the worst possible partition. The palindrome check would require validating that the string itself is equivalent to its own reverse.

For a naive solution, the algorithm would take  $O(n^3)$  time due to the iteration over the input string, each of its substrings, and checking if the substring is a palindrome.

### PROPOSAL

There are various recursive and dynamic programming solutions to this problem.<sup>3</sup> Naive

solutions can take  $O(n^3)$  time while highly optimized solutions can take  $O(n^2)$  time. For my project, I plan to implement both parallel and serial solutions to this problem and compare their performances. With parallelization, I hope to achieve a performance faster than the  $O(n^2)$  serial solutions.

I will compare each solution using various test cases (short string, medium string, long string, very long string, etc.) and metrics such as average, minimum, and maximum time taken. I will also compare using other relevant parallelization metrics such as sparks converted, garbage collected, fizzled, etc.

---

<sup>1</sup> Taken from this Leetcode problem -

<https://leetcode.com/problems/palindrome-partitioning-ii/>

<sup>2</sup> MCM article -

[https://en.wikipedia.org/wiki/Matrix\\_chain\\_multiplication](https://en.wikipedia.org/wiki/Matrix_chain_multiplication)

<sup>3</sup> GeeksForGeeks solutions to the problem -

<https://www.geeksforgeeks.org/palindrome-partitioning-dp-17/>