

Contents

1	June 2021	3
1.1	Longest Palindromic Substring	3
1.2	Container With most Water	3
1.3	Course Schedule	3

Chapter 1

June 2021

1.1 Longest Palindromic Substring

I solved this using dynamic programming.

Basic solution is that:

1. If $\text{str}[i] == \text{str}[j]$, check whether $\text{str}[i+1, j-1]$ is also palindrome.
2. If not, the longest palindrome could be in $\text{str}[i+1, j]$ or $\text{str}[i, j+1]$.

If bottom-up, each cell $\text{dp}[i][j]$ indicates whether $\text{str}[i, j]$ is a palindrome. Deciding whether $\text{str}[i, j]$ is palindrome needs deciding whether $\text{str}[i+1, j+1]$ is palindrome first.

1.2 Container With most Water

Given an array of height.

Find $\min(\text{height}[i], \text{height}[j]) * (j-i)$ such that is maximum.

1.3 Course Schedule

There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array prerequisites where $\text{prerequisites}[i] = [a_i, b_i]$ indicates that you must take course b_i first if you want to take course a_i .

Return true if you can finish all courses. Otherwise, return false.

Example:

A DAG problem