

# Competitive Programming: Beginner to Better (SoC 2025 Midterm Report)

Vijaya Raghavendra S, 23B1042

## Introduction

This document outlines the key topics I have studied so far, along with a list of problems I have solved under each week. For each topic, I have also attached one or two problem solutions that I particularly enjoyed working on.

## 1 Topics Covered

### 1.1 1. Arrays and Strings

**Summary:** Learned about 1D and 2D arrays, character arrays, sliding window, prefix sums, and common string manipulations.

**Problems Solved:**

- Two Sum
- Maximum Subarray (Kadane's Algorithm)
- Longest Substring Without Repeating Characters

**Selected Solution: Longest Substring Without Repeating Characters**

```
def lengthOfLongestSubstring(s):  
    seen = {}  
    start = max_len = 0  
    for i, ch in enumerate(s):  
        if ch in seen and seen[ch] >= start:  
            start = seen[ch] + 1  
        seen[ch] = i  
        max_len = max(max_len, i - start + 1)  
    return max_len
```

### 1.2 2. Linked Lists

**Summary:** Covered singly and doubly linked lists, reversing lists, detecting cycles, and merging lists.

**Problems Solved:**

- Reverse a Linked List
- Detect Cycle in Linked List

- Merge Two Sorted Lists

### Selected Solution: Detect Cycle in Linked List

```
def hasCycle(head):
    slow = fast = head
    while fast and fast.next:
        slow = slow.next
        fast = fast.next.next
        if slow == fast:
            return True
    return False
```

## 1.3 3. Trees and Graphs

**Summary:** Learned about DFS, BFS, binary trees, BSTs, and graph traversal using adjacency lists.

### Problems Solved:

- Inorder Traversal
- Level Order Traversal
- Number of Islands

### Selected Solution: Number of Islands

```
def numIslands(grid):
    def dfs(i, j):
        if i < 0 or j < 0 or i >= len(grid) or j >= len(grid[0]) or grid[i][j] != '1':
            return
        grid[i][j] = '0'
        for x, y in [(0,1), (1,0), (-1,0), (0,-1)]:
            dfs(i+x, j+y)

    count = 0
    for i in range(len(grid)):
        for j in range(len(grid[0])):
            if grid[i][j] == '1':
                dfs(i, j)
                count += 1
    return count
```

## Conclusion

This document summarizes my current understanding and progress across different topics in programming and data structures. I've included a few hand-picked problems that I found particularly interesting and rewarding to solve.