Readings

- Lecture Notes Chapter 7: Divide & Conquer and Recurrence Relations
- Lecture Notes Chapter 13: Stacks & Queues

Problems

Problem 1. Local Maxima
You are given an integer array $A[1..n]$ with the following properties:
- Integers in adjacent positions are different


Example: You have an array $[0, 1, 5, 3, 6, 3, 2]$. There are multiple local maxes at 5 and 6.

Propose an efficient algorithm that will find a local maximum and return its index.

Problem 2. Largest Subarray Sum
Given an integer array (containing both positive and negative values), return the sum of the largest contiguous subarray which has the largest sum.

Problem 3

Given: A binary tree $T$.
Objective: Print the level order traversal of the tree $T$.
Example:

Problem 4

You are given two stacks $S_1$ and $S_2$, each of size $n$.

Implement a queue using $S_1$ and $S_2$. Your queue’s enqueue and dequeue methods should be implemented using only your stacks’ push, pop, and/or peek methods. What are the running times of your new queue’s enqueue and dequeue methods?
Additional Practice Problems

Problem 1. Element Index Matching
You are given a sorted array of \( n \) distinct integers \( A[1...n] \). Design an \( O(\log n) \) time algorithm that either outputs an index \( i \) such that \( A[i] = i \) or correctly states that no such index \( i \) exists.

Problem 2
Given: A binary tree \( T \).
Objective: Print the spiral order traversal of the tree \( T \).
Example:
\[ \text{Hint: Try using 2 stacks.} \]

Problem 3
Given a full stack \( S_1 \) of size \( n \) and an empty stack \( S_2 \) of size \( n \), sort the \( n \) elements in ascending order in \( S_2 \). You may only use the given 2 stacks \( S_1 \) and \( S_2 \) (each of size \( n \)) and \( O(1) \) additional space. What is the running time of your sorting procedure?
Example:
\[ \begin{array}{c|c|c}
4 & & 1 \\
3 & & 2 \\
1 & & 3 \\
5 & & 4 \\
2 & & 5 \\
\end{array} \]
\[ \text{Hint: Start with a simpler example:} \]
Figure 2: *For this tree, your function should print 1, 2, 3, 4, 5, 6, 7.*