UQCS Competitive Programming Group
Week 2
Data Structures

Matthew Low

UQ Computing Society

Welcome to CPG!

UQCS **Competitive Programming Group**.

Algorithms, data structures and, most of all, problem solving.

We will try to be beginner-friendly, but basic programming knowledge is expected. COMP3506 (or any algorithms course) would also be helpful for tackling these problems.

Last week!

We got used to LeetCode, and tried out some introductory questions (including one that was too hard. . .)

Median of Two Sorted Arrays

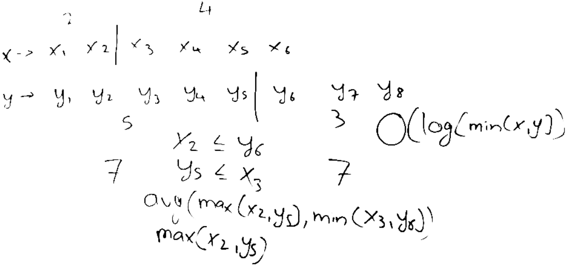


Figure 1: Basic idea.

I am indebted to this video: `https://www.youtube.com/watch?v=LPFhl65R7ww` explains it much better than I ever could. Python code on `https://github.com/UQComputingSociety/cpg`.

This week!

We'll be covering **three** problems dealing with **data structures**.

Most of this material is based off material in *Competitive Programming 3*. We'll cover three fundamental data structures, and one related problem from each.

- Stacks
- Trees
- 2D arrays

You are highly encouraged to try out more of these problems for yourself! Look in CP3 for more.

To those wanting more of a challenge: harder problems will come once we've covered the data structures underlying them. :)

Want more specifics?

- COMP3506: Algorithms and Data Structures
- COMP4500: *Advanced* Algorithms and Data Structures

We'll just be doing problems here. If you haven't done either (or are doing them at the moment), don't worry.

Trees

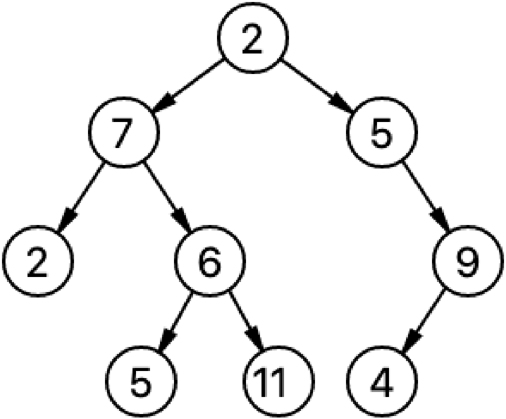


Figure 2: `https://en.wikipedia.org/wiki/Binary_tree`

We particularly focus on **binary trees**.

Invert a binary tree

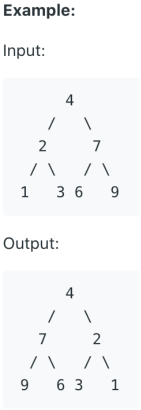
> *Google: 90% of our engineers use the software you wrote (Homebrew), but you can't invert a binary tree on a whiteboard so f*** off.*

Max Howell `@mxcl`, Homebrew developer.

. . . so do you wanna be hired at Google?

Problem 1: Invert Binary Tree

Invert a binary tree.



Example:

Input:

```
     4
   /   \
  2     7
 / \   / \
1   3 6   9
```

Output:

```
     4
   /   \
  7     2
 / \   / \
9   6 3   1
```

Figure 3: `https://leetcode.com/problems/invert-binary-tree/`

Stacks

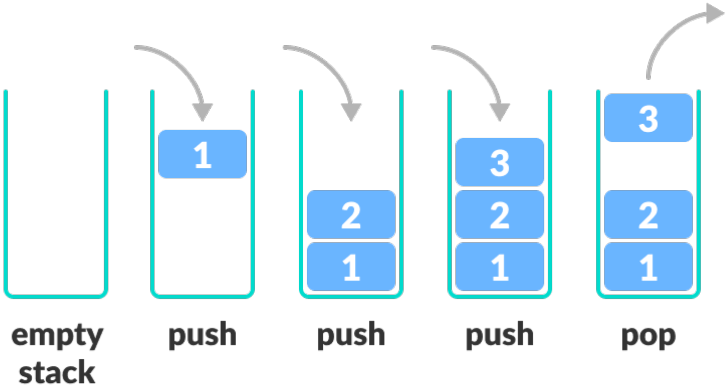


Figure 4: `https://www.programiz.com/dsa/stack`

Problem 2: Basic Calculator II

Implement a basic calculator to evaluate a simple expression string.

The expression string contains only **non-negative** integers, $+$, -, *, / operators and empty spaces. The integer division should truncate toward zero.

Example 1:

```
Input: "3+2*2"
Output: 7
```

Example 2:

```
Input: "3/2"
Output: 1
```

https://leetcode.com/problems/basic-calculator-ii/

2D arrays

Arrays. . . but 2D.

Problem 3: Valid Sudoku

Determine if a 9x9 Sudoku board is valid. Only the filled cells need to be validated according to the following rules:

Each row must contain the digits 1-9 without repetition. Each column must contain the digits 1-9 without repetition. Each of the 9 3x3 sub-boxes of the grid must contain the digits 1-9 without repetition.

Notes:

- A Sudoku board (partially filled) could be valid but is not necessarily solvable.
- Only the **filled cells** need to be validated according to the mentioned rules.
- The given board contain only digits `1-9` and the character `'.'`.
- The given board size is always `9x9`.

`https://leetcode.com/problems/valid-sudoku/`

Challenge (discuss on #cpg)

How about *solving* a Sudoku puzzle?

`https://leetcode.com/problems/sudoku-solver/`