

UQCS Competitive Programming Group

Week 1

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.

Textbook

The “text” is *Competitive Programming 3*. We will **not** (at least for now) be using problems out of this book, but we will be following the contents as a rough guide of what to cover. For more problems, refer to CP3, there are hundreds in each section! However, note that this book has code snippets written in C/C++/Java. The reference for each section is given in the “CP3 reference” column of the schedule at the bottom of this README.

A rough schedule

#	Starting	Topic	CP3 reference
1	3 Aug	Beginnings	N/A
2	10 Aug	Data Structures	2.1-2.4
3	17 Aug	Complete Search	3.1-3.2
4	24 Aug	Divide & Conquer and Greedy	3.3-3.4
5	31 Aug	Dynamic Programming	3.5
6	7 Sep	String Processing	6.1-6.4
7	14 Sep	DFS, BFS and Graphs	4.1-4.2
8	21 Sep	MST/Shortest Paths	4.3-4.5
—	28 Sep		
9	5 Oct	Network Flows	4.6
10	12 Oct	Mathematics	5.4-5.6
11	19 Oct	Computational Geometry	7.1-7.3
12	26 Oct	...	

This is a group, not a lecture.

Please suggest problems!

Please suggest helpful resources!

Please suggest improvements for future weeks!

This is the first time UQCS is doing anything like this, so any feedback and assistance would be much appreciated.

Online or in-person?

Online for the near future. As a result, things will be a bit different.

We'll put up problems to solve and a few of us will be available to help out with any difficulties. Just unmute and talk in Zoom, or message in the chat.

This is a collaborative learning group, so after the first few weeks or so, there will be no “instructors” as such. We'll all be helping each other on!

Competitive programming...competition?

Yes! ICPC South Pacific Regional Contest, October 17
(COVID-permitting). More details to come soon (hopefully).

Problem 1: Two Sum

Given an array of integers, return indices of the two numbers such that they add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice.

Problem 2: Letter Combinations of a Phone Number

Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent.

A mapping of digit to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.



Problem 3: Median of Two Sorted Arrays

There are two sorted arrays `nums1` and `nums2` of size m and n respectively. Find the median of the two sorted arrays. The overall run time complexity should be $\mathcal{O}(\log(m + n))$. You may assume `nums1` and `nums2` cannot be both empty.