The University of Melbourne
School of Computing and Information Systems
COMP90038 Algorithms and Complexity

# Assignment 2, Semester 2, 2018

Released: Tuesday 25 September. Deadline: Sunday 14 October at 23:59

## Objectives

To improve your understanding of data structures and algorithms for sorting and search. To consolidate your knowledge of trees and tree-based algorithms. To develop problem-solving and design skills. To develop skills in analysis and formal reasoning about complex concepts. To improve written communication skills; in particular the ability to use pseudo-code and present algorithms clearly, precisely and unambiguously.

## Problems

1. [1 Mark] Consider the data sequence $S = [82,\ 91,\ 13,\ 92,\ 64,\ 10,\ 28,\ 55,\ 96,\ 97]$. Draw a valid AVL tree for it, assuming that the data has arrived one at the time. Show detailed steps by giving the AVL tree after inserting each element.

2. Consider two sets of integers, $X = [x_1,\ x_2,\ \ldots,\ x_n]$ and $Y = [y_1,\ y_2,\ \ldots,\ y_n]$. Write two versions of a FindSetIntersection$(X, Y)$ algorithm to find the common elements in both sets. Each of your algorithms should return an array with the common elements, or an empty array if there are no common elements.

   You may make use of any algorithm introduced in the lectures to help you develop your solution. That is, you do not have to write the 'standard' algorithms – just use them. Therefore, you should be able to write each algorithm in about 10 lines of code. **You must include appropriate comments in your pseudocode.**

   (a) [2 Marks] Write a pre-sorting based algorithm of FindSetIntersection$(X, Y)$. Your algorithm should strictly run in $\mathcal{O}(n \log n)$.

   (b) [2 Marks] Write a Hashing based algorithm of FindSetIntersection$(X, Y)$. Your algorithm should run in $\mathcal{O}(n)$.

3. [4 Marks] Sloppy Inc. has a very unusual way to communicate the decisions made by its CEO to all employees. Each day, any employee that knows the decision can disclose it to at most one of its direct subordinates. Design an efficient algorithm to compute the minimum number of days required for the decision to be disclosed to all employees. What is the time complexity of your algorithm?

   To help you design your algorithm, assume that Sloppy Inc. has a hierarchical structure resembling an n-ary tree. Each employee is labeled $\{0, 1, \ldots, n-1\}$, where 0 corresponds to root of the tree (the CEO). To store the tree you can use a two-dimensional array $C[n][n]$, where $k = C[i][0]$ is the number of direct subordinates of employee $i$, and $C[i][1 \ldots k]$ contains the labels of each direct subordinate employee. Any other entry in the array has value of $-1$. Note that the order in which the messages are distributed matters, e.g., employees with deeper subordinate trees should probably receive the message first.

   *Hint*: Solving this problem requires a recursion.

4. Given an array of $n$ numbers $A[0 \ldots n-1]$. Write an efficient algorithm for below cases:

    (a) [3 Marks] For each of the element $A[i]$, find the minimum $j$ so that $A[j] > A[i]$ and $j > i$. Your algorithm should return an array of length $n$. If such $j$ does not exist for some $i$, that entry should be -1. What is the complexity of your algorithm?

    index

    (b) [3 Marks] For each of the element $A[i]$, find the minimum $A[j]$ so that $A[j] > A[i]$ and $j > i$. Your algorithm should return an array of length $n$. If such $j$ does not exist for some $i$, that entry should be -1. Your algorithm should have $\mathcal{O}(n \log n)$ complexity.

    To help you verify your algorithm, for the sequence [80, 19, 49, 45, 65, 71, 76, 28, 68, 66] the results are:

    (a) $[-1,\ 2,\ 4,\ 4,\ 5,\ 6,\ -1,\ 8,\ -1,\ -1]$

    (b) $[-1,\ 7,\ 4,\ 4,\ 9,\ 6,\ -1,\ 9,\ -1,\ -1]$

## Submission and Evaluation

- You must submit a PDF document via the LMS. Note: handwritten, scanned images, and/or Microsoft Word submissions are not acceptable — if you use Word, create a PDF version for submission.

- Marks are primarily allocated for correctness, but elegance of algorithms and how clearly you communicate your thinking will also be taken into account. Where indicated, the complexity of algorithms also matters.

- We expect your work to be neat — parts of your submission that are difficult to read or decipher will be deemed incorrect. Make sure that you have enough time towards the end of the assignment to present your solutions carefully. Time you put in early will usually turn out to be more productive than a last-minute effort.

- You are reminded that your submission for this assignment is to be your own individual work. For many students, discussions with friends will form a natural part of the undertaking of the assignment work. However, it is still an individual task. You should not share your answers (even draft solutions) with other students. Do not post solutions (or even partial solutions) on social media or the discussion board. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student concerned.

    Please see `https://academicintegrity.unimelb.edu.au`

If you have any questions, you are welcome to post them on the LMS discussion board *so long as you do not reveal details about your own solutions.* You can also email the Head Tutor, Lianglu Pan (lianglu.pan@unimelb.edu.au) or the Lecturer, Andres Munoz-Acosta (munoz.m@unimelb.edu.au). In your message, make sure you include COMP90038 in the subject line. In the body of your message, include a precise description of the problem.

## Late Submission and Extension

Late submission will be possible, but a late submission penalty will apply: a flagfall of 2 marks, and then 1 mark per 12 hours late. Extensions will only be awarded in extreme/emergency cases, assuming appropriate documentation is provided simply submitting a medical certificate on the due date will not result in an extension.