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

Assignment 1, Semester 1 2020

Deadline: Friday April 24, 9:00am 30 marks (15% of final assessment)

Objectives

To improve your understanding of the time complexity of algorithms and recurrence relations. To develop problem-solving and design skills. To improve written communication skills; in particular the ability to present algorithms clearly, precisely and unambiguously.

Problems

1. **[4 marks]** Consider the following pseudocode:

```
$x \leftarrow 0$
for $k \leftarrow 1$ to $n$ do
    for $j \leftarrow k$ to $2n$ do
        $x \leftarrow x + 1$
```

   (a) How many times is the inner-most statement ($x \leftarrow x+1$) executed? Justify your answer using the summation notation and rules as described in Appendix 1 of Levitin.

   (b) Express the time complexity of the code segment using the appropriate Big $O/\Omega/\Theta$ notation. That is, you should make the strongest possible claim about the complexity of the code segment.

2. **[4 marks]** Consider the following recurrence relation:

$$T(n) = \begin{cases} 0 & \text{if } n = 1 \\ 3T(n/3) + 2cn & \text{otherwise} \end{cases}$$

   where c is a positive constant. Assume that $n = 3^m$.

   Solve the recurrence. Your answer should show how you derive the closed form.

3. **[8 marks]** A manager wants to purchase **two** different products from a list of n products with a fixed budget k. Assume the prices of the n products are distinct, which are stored in an array A sorted in increasing order of price.

   Design an algorithm to find how many different combinations that the manager can choose from. For example, if there are five different products, of which the prices are $A = [1, 2, 3, 4, 6]$ the budget $k = 4$, the algorithm should return 2 as only the combinations of $(1, 2)$ and $(1, 3)$ do not exceed the budget.

   Note, that the algorithm would be incorrect if it returned a value of 1 corresponding to the product in $A[3] = 4$ as this would not meet the criteria of 'two different products' given the budget constraint.

   Full marks will be given if your algorithm runs in $O\big(n \log(n)\big)$ time. Your algorithm should be presented in unambiguous pseudocode or Python code.

4. **[4 marks]** Your colleague has partially implemented an algorithm to test if a strongly connected graph is bipartite (can be 2-coloured). Pseudocode for her algorithm is presented below.

   Your colleague has won the lottery and decided to leave university, so it is up to you to finish the algorithm. Unfortunately, she did not define the data structure for the variable X. However, you can tell from the documentation that the data structure was supposed to be either a *stack* or a *queue*.

```
function IsBipartite(G[0...n − 1][0...n − 1])      ▷ Input is an adjacency matrix n × n
    numSeen = 0
    colour[0...n-1] ← [-1, ···, -1]
    X ← EmptyQueueOrStack( )
    PushOrEnqueue(X, <0, RED>)
    while numSeen < n do
        i, c ← PopOrDequeue(X)
        if colour[i] = -1 then
            numSeen ← numSeen + 1
        colour[i] ← c
        nextColour ← RED
        if c = RED then
            nextColour ← BLUE
        for j ← 0 to n − 1 do
            if G[i][j] = 1 then
                if colour[j] = c then
                    return false
                PushOrEnqueue(X, <j, nextColour>)
    return true
```

   (a) Will a *stack* data structure or *queue* data structure produce the correct answer?

   (b) Provide an adjacency matrix for a small graph that causes the other data structure to either return the wrong result or not terminate. (Hint: a 3 node graph is sufficient)

5. **[10 marks]** You are an employee of The Department of Environments and Fire Assessment. You have been assigned a task to write an algorithm to calculate the size of the *largest connected burnt area* in the state of Victoria as a result of the devastating recent bushfires.

   A number of simplifying assumptions have been made in this task:

   - We will represent the environment using a 2D array (or an abstract map of the area under consideration). Here, the actual scale is not relevant in the problem formulation.
   - Each cell in the 2D array contains a binary variable indicating whether the cell has been burnt or not. For example the value of cell $c_{i,j} = 1$ if the area represented by that site has been burnt. In contrast, the value of cell $c_{i,j} = 0$ if the area represented by that site has **not** been burnt
   - A 'connected burnt area' is defined as a collection of connected cells where $c_{i,j} = 1$ (ie., burnt) for all cells. Two cells $c_{m,n}$ and $c_{x,y}$ are connected if either of the cells is a north, south, east, or west neighbour of the other.

   For example, in the 2D array below, the size of the largest connected burnt area is 5.

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

   Your algorithm should be presented in unambiguous pseudocode or Python code. *Note*: it may be necessary to modularise your solution using multiple functions.

   Full marks will only be awarded if your algorithm is correct with appropriate time complexity. Here, we have not provided a clear definition of 'appropriate time complexity'. We do this, as we want you to think carefully about the design of your solution – lots and lots of nested for loops is not the correct approach to use.

   Hints:

   - For each cell $c_{i,j}$, determine which of the other cells it is connect to.
   - How could a graph traversal algorithm be used in this context?

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.
- Please write any pseudocode following the format suggested in the examples provided in the sample lecture slides and/or the textbook. Take care with indentation, loops, if statements, initialisation of variables and return statements. **Python code is acceptable**.
- 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. 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. You can also email the Head Tutor, Lianglu Pan <`lianglu.pan@unimelb.edu.au`> or the Lecturer, Michael Kirley <`mkirley@unimelb.edu.au`>. In your message, make sure you include COMP90038 in the subject header. In the body of your message, include a precise description of the problem.

Extension policy: obviously COVID-19 has impacted on all students. We have carefully taken this issue into consideration when designing the questions and the time window available to attempt the assignment. It is in your best interest to complete the assignment by the due date so that there is ample time to complete the remaining assessment tasks in this subject.

Late submission will be possible, however **a late submission penalty of 2 marks per day may apply**.