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

Assignment 1, Semester 1 2019

Deadline: Monday April 15, 9:00am   Marks available: 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]**

   Use repeated substitution (or telescoping) to determine the asymptotic upper bound for the running time of an algorithm defined by the recurrence relation below:

   $$f(n) = 4f(n/2) + g(n) \qquad \text{with } g(n) \in \Theta(n) \qquad \text{and } f(1) \in \Theta(1)$$

2. **[8 marks]**

   Given an array $A[.]$ of n integers, design an efficient algorithm to calculate the *maximum product* of any $n-1$ elements in the array.

   Your algorithm should be presented in unambiguous pseudocode using the template below. Full marks will only be awarded if your algorithm is correct and has time complexity $O(n)$.

   function MaxProduct($A[.], n$)   ▷ input: an integer array A of size n

   /* *fill in the details* */

   return maxAnswer

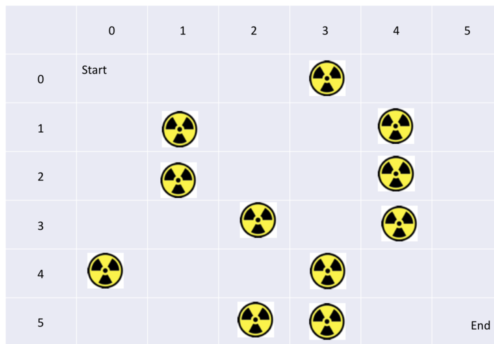
3. **[6 marks]**

   As part of the development of the new student precinct at the University of Melbourne, an old single storey building previously used for the storage of radioactive isotopes was scheduled to be demolished. Unfortunately, higher than acceptable levels of radiation (values above a threshold level) were observed in some parts of the building. Consequently, the demolition contractors developed a $n \times n$ grid or map to record the relative radiation levels on the building floor plan.

   Specifically, they used a two-dimensional binary array $A[0,..,n-1][0,..,n-1]$, where a value $A[i][j] = 1$ indicated a radiation level above the acceptable threshold level, and a value of $A[i][j] = 0$ indicated 'safe' locations, that is, where the recorded radiation level was below the threshold value.

One of the contractors had to navigate their way across the building, starting at position $A[0][0]$ and ending at position $A[n-1][n-1]$. Given the delicate nature of this work, the contractor could only move one cell at a time, by either moving one cell to the right or one cell down. They also wanted to minimize the number of cells that they moved through where the recorded radiation level was above the acceptable threshold level (ie., where $A[i][j] = 1$).

Write a recursive function to determine the minimum number of cells that the contractor must pass through where the radiation level was above the acceptable threshold level. For the example below (where the radiation symbol indicates $A[i][j] = 1$), your algorithm should return a value of 1.



4. **[8 marks]**

   Consider an unweighted, undirected graph $G = \langle V, E \rangle$. The *neighbourhood* of a node $v \in V$ in the graph is the set of all nodes that are adjacent (or directly connected) to v. Subsequently, we can define the *neighbourhood degree* of the node v as the sum of the degrees of all its neighbours (those nodes that are directly connects to v).

   (a) Design an algorithm that returns a list containing the *neighbourhood degree* for each node $v \in V$, assuming that the input is the graph, G, represented using an adjacency list. Each item i in the list that you generate will correspond to the correct value for the *neighbourhood degree* of node v_i.

   Your algorithm should be presented in unambiguous pseudocode. Your algorithm should have a time complexity value $O(V + E)$.

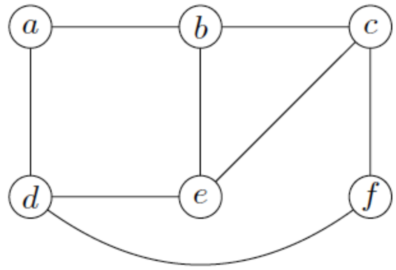
   (b) If an adjacency matrix was used to represent the graph instead of an adjacency list, what is the new value for the time complexity? Justify your answer by explicitly referring to the changes that would be necessary to your algorithm from part (a).

5. **[4 marks]**

   Suppose we have $2n$ people, some of which are related to some of the others. We might want to split them into groups of two, so that the two people in a group are related (if this is possible).

   Expressing this as a graph problem, suppose we have an undirected graph $G = \langle V, E \rangle$. A *pairing* is a set $P \subseteq E$ of edges such that for all $(u, v), (x, y) \in P$, the nodes u, v, x, y are all different. In other words, no two edges in P have a node in common. A *complete pairing* is a pairing P that uses all the graph's nodes, that is, a pairing for which $\bigcup_{(u,v)\in P}\{u, v\} = V$.

For example, in the graph shown below, one possible complete pairing is $\{(a,b),(d,e),(c,f)\}$, but there is no complete pairing that contains the edge (b,c).



Use mathematical induction to show how to construct a graph with $2n$ nodes and n^2 edges such that the graph has exactly one, unique, complete pairing.

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. Cormen et al. (available as an e-book in the library) provide some guidelines for pseudocode (pages 20–22).
- 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.

Late submission will be possible, but **a late submission penalty will apply**: a flagfall of 4 marks, and then 2 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.