THE UNIVERSITY OF SYDNEY

CONFIDENTIAL EXAM PAPER

This paper is not to be removed from the exam venue.

Computer Science

EXAMINATION

Semester 2- Main, 2022

COMP9123 Data Structures and Algorithms

EXAM WRITING TIME: 2 hours
READING TIME: 10 minutes

EXAM CONDITIONS:
This is a OPEN book examination.
All submitted work must be **done individually** without consulting someone else's help, in accordance with the University's "Academic Dishonesty and Plagiarism" policies.

MATERIALS PERMITTED IN THE EXAM VENUE:

MATERIALS TO BE SUPPLIED TO STUDENTS:

INSTRUCTIONS TO STUDENTS:
Type your answers in your text editor (Latex, Word, etc.) and convert it into a pdf file.

Submit this pdf file via Canvas. No other file format will be accepted. Handwritten responses will **not** be accepted.

Start by typing you student ID at the top of the first page of your submission. Do **not** type your name.

Submit only your answers to the questions. Do **not** copy the questions.

Do **not** copy any text from the permitted materials. Always write your answers in your own words.

For examiner use only:

Problem	1	2	3	4	Total
Marks					
Out of	10	10	20	20	60

Problem 1.

a) Analyse the time complexity of this algorithm as a function of n, the size of the input array A. [4 marks]

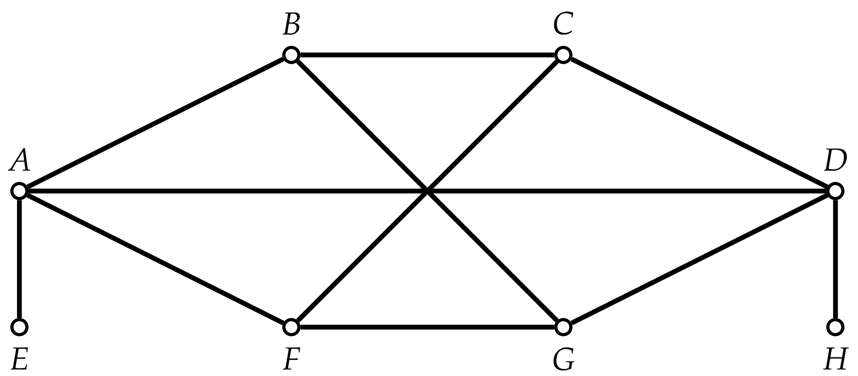
```
def Process(A)
    t ← 0
    for i = 0; i < n; i++ do
        if A[i] ≤ t * t then
            t ← t + 1
    return t
```

b) Solve the following recursion: [3 marks]

$$T(n) = \begin{cases} 2T(n/3) + O(n) & \text{for } n > 1 \\ O(1) & \text{for } n = 1 \end{cases}$$

c) We have seen in class several sorting algorithms with time complexity $O(n \log n)$, where n is the size of the input array (number of elements to sort). **True or False:** if the input array is promised to only contain integers between 0 and 10, there exists an algorithm to sort it in time $O(n)$. *(No justification needed.)* [3 marks]

Problem 2. Consider the following undirected graph:



Assuming that ties are broken alphabetically, **your task** is to:

a) Compute the breadth-first search tree T of the graph starting from A. List the edges in T. [5 marks]

b) Compute the depth-first search tree T of the graph starting from H. List the edges in T. [5 marks]

(You do **not** have to explain your answer.)

Problem 3. We are given as input an array A of $4n$ (distinct) integers, and want to partition these $4n$ numbers into 4 arrays A_1, A_2, A_3, A_4, each of size n, such that all numbers in A_1 are smaller than all numbers in A_2, all numbers in A_2 are smaller than all numbers in A_3, and all numbers in A_3 are smaller than all numbers in A_4.

Your task is to give an algorithm for this task running in time $O(n)$.

a) Describe your algorithm in plain English. [5 marks]

b) Analyze the time complexity of your algorithm and briefly argue its correctness. [5 marks]

c) Generalise to the case where, instead of 4 arrays, you must output k of them (where k is an input parameter such that n is divisible by k) where all numbers in array A_i are smaller than all numbers in A_j whenever $1 \leq i < j \leq k$. Describe an algorithm for this task with time complexity $O(kn)$, and briefly argue its correctness and time complexity. *(Note that k can be anything between 1 and n, and is not considered a constant.)* [5 marks]

d) Give another algorithm for the k-array variant with time complexity $O(n \log n)$, and briefly argue its correctness and time complexity. When is it better to use the one from question c), and when is it better to use this one? [5 marks]

Problem 4. You are employed at a chemical facility, and are put in charge of a new project to optimise the production of molecules. Your lab has access to specific molecules, and can use a small set of techniques to transform them into different ones, through chemical reactions. Specifically, there is a set of n known molecules, and 3 possible chemical reactions you can apply to each of them. Your goal is to find out the most cost-effective way, given a starting molecule s and a target molecule t, to use a chain of reactions to transform s into t.

To do so, you are given the set of molecules and transformations as a *directed* graph $G = (V, E)$, where V corresponds to the set of n molecules, and an edge $(u, v) \in E$ indicates that there exists a chemical reaction transforming u into v. *(Keep in mind that there are only 3 possible reactions you can apply to a molecule!)* Moreover, each edge $(u, v) \in E$ comes with an positive integer c_{uv}, corresponding to the cost of the reaction. A sequence of successive chemical reactions then has total cost equal to the individual costs of each reaction. You are provided this graph in the adjacency list representation.

Your task is to give three algorithms to solve the tasks below: briefly describe the algorithms in plain English, and (briefly) prove their correctness and time complexity.

a) On input G, a starting molecule $s \in V$, and a target molecule $t \in V$, output whether there exists a sequence of chemical reactions allowing you to transform s into t. This must run in time $O(n)$. [6 marks]

b) On input G, an integer $k \geq 1$, a starting molecule $s \in V$, and a target molecule $t \in V$, output whether there exists a sequence of at most k successive chemical reactions allowing you to transform s into t. This must run in time $O(n)$. [7 marks]

c) On input G, a starting molecule $s \in V$, and a target molecule $t \in V$, output a cheapest sequence of chemical reactions allowing you to transform s into t (if there is one); if there is none, output impossible. This must run in time $O(n \log n)$. [7 marks]