National University of Singapore

School of Computing

CS2040C - Data Structures and Algorithms

Final Assessment

(Semester 1 AY2019/20)

Time Allowed: 2 hours

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this assessment paper until you are told to do so.
2. This assessment paper contains THREE (3) sections.
   It comprises FOURTEEN (14) printed pages, including this page.
3. This is an **Open Book Assessment**.
4. For Section A, answer **ALL** questions using the OCR form provided.
   For Section B and C, answer **ALL** questions within the **boxed space** in this booklet.
   Only if you need more space, then you can use the empty page 14.
   You can use either pen or pencil. Just make sure that you write **legibly**!
5. Important tips: Pace yourself! Do **not** spend too much time on one (hard) question.
   Read all the questions first! Some questions might be easier than they appear.
6. You can use **pseudo-code** in your answer but beware of penalty marks for **ambiguous answer**.
   You can use **standard, non-modified** algorithm discussed in class by just mentioning its name.
7. Please write your Student Number only. Do **NOT** write your name.

A	0							

This portion is for examiner's use only

Section	Maximum Marks	Your Marks	Remarks
A	50		
B	20		
C	30		
Total	100		

A MCQs (50 marks)

Select the **best unique** answer for each question.
Each correct answer worth 2 marks but each **wrong answer** worth -1 mark.

The MCQ section is not supposed to be archived to open up possibilities of reuse in the future.
PS: Your score for this section will not be lower than 0.

B Create Test Cases (20 marks)

Create a test case for each scenario below. Each valid test case worth 4 marks.
However, if if is not possible to create any test case to satisfy the requirement, explain the reason.

1. Draw an AVL Tree with $N = 11$ Integers (numbered with $[1..N]$) such that it's height is 4. Recall that we define height to be the number of edges from the root to its deepest leaf.

2. Draw a **connected** undirected simple graph with **exactly 7** vertices (labeled with [0..6]) and **at least 6** undirected edges such that both DFS(0) – Depth-First Search from source vertex 0 – visits only four vertices: {0, 2, 4, 6}.

3. Draw an undirected unweighted simple graph with **exactly 7** vertices (labeled with [0..6]) and **exactly 11** undirected edges such that there are **exactly 3** Connected Components.

4. Draw a simple Directed Acyclic Graph (DAG) with **exactly 7** vertices (labeled with [0..6]) and **at least 7** directed edges such that *any* topological finding algorithm (DFS variant, Kahn's/modified BFS variant, etc) will report exactly **the same** topological ordering in this DAG.

5. Draw a simple Directed weighted graph with **exactly 7** vertices (labeled with [0..6]) and **exactly 7** directed distinct weighted edges (possibly weights are 1, 2, 3, 4, 5, 6, 7) such that the shortest paths from source vertex $s = 0$ to the other 6 vertices are 1/2/3/4/5/6, respectively.

Section B Marks = (___+___)+(___+___+___) = _____

C Applications (30 marks)

Imagine a (fictional) country being invaded by its enemy. The enemy troops are establishing their bases all over your country. You are only safe on the towns that are 'sufficiently far away' from all current enemy bases. You need to quickly write a program to help you determine where to move.

In the first line, you will be given four integers N, M, E, and K ($1 \leq N \leq 10\,000$, $0 \leq M \leq 100\,000$, details of E is in the subtask below, and $1 \leq K \leq 100$), giving the number of towns in your country, the number of roads between them, the number of bases that the enemies are going to build *one by one*, and the minimum safe distance from those enemy bases, respectively. The towns in your country are assigned numbers 1, 2, ..., N. You can assume that your country is a connected graph.

The following M lines describe the *bidirectional* roads; each of them contains three integers $T1$, $T2$ ($1 \leq T1 < T2 \leq N$) and D (details of D is in the subtask below), where D is the length of the road between towns T1 and T2. There is at most one direct road between any two towns.

The last line contains E integers that describe the positions of enemy bases; the i-th of them contains the number Bi ($1 \leq Bi \leq N$) of the town where the enemies build their i-th base.

Your job is to output E lines. On i-th output line, write the number of towns that are safe after the enemies build their i-th base. The town is safe if its distance from *any* of the towns $B1$, $B2$, ..., Bi (that have enemy base built at step i) is **at least** K.

Sample Input/Output and One Other Similar Test Case (4 marks)

Observe Sample Input 1 (see Figure 1) and the given Output 1. If you have understood this question, write the output of Input 2 (same graph, just different K and E enemy bases) in the box below!

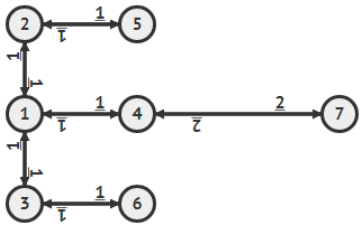

Figure 1: The graph of Sample Input 1

Input 1	Output 1	Input 2	Output 2 (4 marks)
7 6 4 3	2	7 6 4 2 (K = 2)	
1 2 1	1	1 2 1	
1 3 1	0	1 3 1	
2 5 1	0	2 5 1	
3 6 1		3 6 1	
1 4 1		1 4 1	
4 7 2		4 7 2	^ write your
2 1 4 3		7 4 1 2 <<-----	^ answer above

Graph Data Structure (4 marks)

How are you going to store the graph?
Describe the details of your chosen graph data structure

Subtask 1 (5 marks)

Let $E == 1$ and $D == 1$.

What are the significance of those constraints? (1 mark)

What is your proposed algorithm to solve this Subtask 1 (in pseudo-code)? (3 marks)

What is the time complexity of your proposed algorithm above in terms of N and M? (1 mark)

Subtask 2 (5 marks)

Let $E == 1$ (same as Subtask 1) but $1 \leq D \leq 100$.

What is the significance of the updated constraints? (1 mark)

What is your proposed algorithm to solve this Subtask 2 (in pseudo-code)? (3 marks)

What is the time complexity of your proposed algorithm above in terms of N and M? (1 mark)

Subtask 3 (12 marks)

Let $1 \leq E \leq N$ and $1 \leq D \leq 100$ (same as Subtask 2).

What is the significance of the updated constraints? (1 mark)

What is your proposed algorithm to solve this Subtask 3 (in pseudo-code)? (7 marks)

What is the time complexity of your proposed algorithm above in terms of N and M?
Notice that $E = O(N)$. So you need to ensure that your algorithm above executes not more than 10^8 operations even with $N = 10\,000$ and $M = 100\,000$. (4 marks)

Section C Marks = ___+___+___+___+___ = _____

– End of this Paper, All the Best –