National University of Singapore
School of Computing

CS2010 - Data Structures and Algorithms II

(Semester 4: AY2015/16)

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 TEN (10) printed pages, including this page.
3. This is an **Open Book Assessment**.
4. Answer **ALL** questions within the **boxed space** in this booklet.
   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. Write your Student Number in the box below:

A	0	1						

This portion is for examiner's use only

Section	Maximum Marks	Your Marks	Remarks
A	20		
B	15		
C	65		
Total	100		

A Basics (20 marks)

Please fill in your answers on the blank spaces provided. You are given one example per each question.

1. **(4 marks)** List down **four** more *non linear* data structures.

   (a) Adjacency Matrix

   (b)

   (c)

   (d)

   (e)

2. **(4 marks)** List down **four** more *graph* algorithms and the graph problems that they solve.

   (a) Depth-First Search: An algorithm for traversing a graph,

   (b)

   (c)

   (d)

   (e)

3. **(7 marks)** List down **7** more data structure/algorithm *abbreviations* and their long forms.

   (a) DFS: Depth-First Search,

   (b)

   (c)

   (d)

   (e)

(f)

(g)

(h)

4. **(2 marks)** List down **2** more algorithms that uses ADT Priority Queue.

   (a) Solution of CS2010 PS1 of this S4 AY2015/16: Scheduling Deliveries Ultimate,

   (b)

   (c)

5. **(3 marks)** List down **3** more graph algorithms that have $O(V + E)$ time complexity.

   (a) DFS: Depth-First Search,

   (b)

   (c)

   (d)

Section A Marks = _____

B Analysis (15 marks)

Prove (the statement is correct) or disprove (the statement is wrong) the statements below.

1. Inserting a new element that is greater than the maximum element currently in a Binary Max Heap with $N > 15$ elements will surely trigger more than 3 swaps.

2. A balanced Binary Search Tree is the best data structure to implement efficient standard First-In First-Out (FIFO) Queue (*not* Priority Queue).

3. If a connected graph has > 0 articulation point (cut vertex), then it must have > 0 bridge too.

4. If you run Prim's algorithm on a connected weighted undirected graph with $V > 3$ vertices and $E > 2$ edges from *any* source vertex, then it will *always* take *the first 2* edges dequeued from its Priority Queue into the Minimum Spanning Tree (MST).

5. Running Depth First Search (DFS) algorithm to solve the Single-Source Shortest Paths (SSSP) problem on any non Tree graph will *always* produce wrong answers.

Section B Marks = _____

C Applications (65 marks)

C.1 Organizational Chart (20 marks)

You are one of the employee in a super gigantic software company consisting of N ($1 < N \leq 100\,000$) different person and owned by a rich family. This rich family consists of F ($1 \leq F \leq N$) family members and each of them handles a division in the company. In this company, each division has a top-down and acyclic chain of commands (that looks like a *directed tree*), e.g. each owner (who does not need to report to anybody) manages several bosses who report directly only to him/her, each boss manages several managers who report directly only to him/her, each manager manages several minions who report directly only to him/her, and so on until we reach a certain job level where that employee does not manage anybody else (note that the structure can be quite deep). In this company, each division works independently and do not interfere with each other.

For example, in Figure 1, we see an *example* organizational chart of this company. Each person in the company is given an id(entifier number) between [0..N-1]. Your id is X. The arrow directions in Figure 1 show the reporting structure in this company.

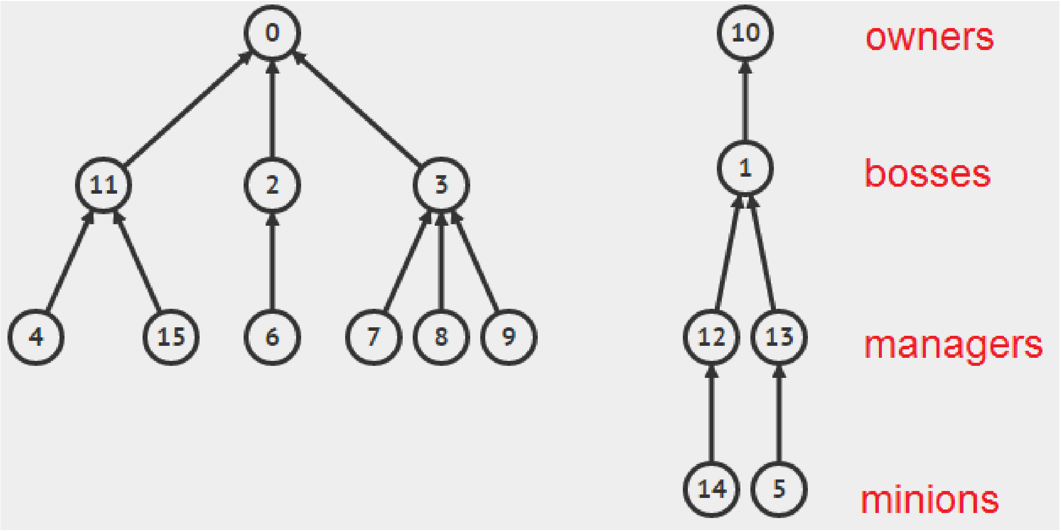

Figure 1: A Sample Organizational Chart with $F = 2$ (that is, 2 divisions).

You want to answer the following self-reflection questions about this company:

Subtask 1: Data Structure (5 marks)

What is the best (graph) data structure to store such organization chart like in Figure 1 and what is the space complexity?

Subtask 2: List the Owner(s) (5 marks)

After you store the organizational chart into your chosen data structure in previous Subtask 1, design the best algorithm to output the list of owner(s) in this company and analyze its time complexity.

If given Figure 1 as input, the answer should be 2 ids: Person with ids $\{0, 10\}$.

Subtask 3: List the Colleague(s) (10 marks)

Given your id X, design the best algorithm to output the list the colleague(s): employee(s) with **the same job title as you in your own division** and analyze its time complexity. You have to use the data structure in previous Subtask 1. List your colleague(s) according to their increasing ids.

For the purpose of this question, your id X will never be one of the answer in previous Subtask 2. Job title is defined as the 'distance from owner', e.g. If given Figure 1 as input and your id is:

- $X = 2$, a boss, then you have to answer 2 other bosses ids $\{3, 11\}$ as all of these ids have 1 unit distance from the owner(s) (boss with id 1 is in different division),
- $X = 7$, a manager, then you have to answer 5 other manager ids $\{4, 6, 8, 9, 15\}$ as all these ids have 2 unit distance from the owner(s) (managers with ids $\{12, 13\}$ are in different division),
- $X = 14$, a minion, then you have to answer 1 other minion id $\{5\}$ as all these ids have 3 unit distance from the owner(s), and so on.

Section C.1 Marks = _____

C.2 Cost Saving (15 marks)

Currently at night, all lights along all corridors between NUS buildings are turned on while only a few students pulling all-nighters (and security guards) plying through those corridors at night.

Electricity is a precious resource and you are tasked to perform cost saving on this matter by strategically turning off some of these lights *yet still providing those pulling all-nighters at least one safe – that is, iluminated – path between any two buildings in NUS*. For this question, you are given a weighted connected undirected graph that describes the (integer) cost of turning on the lights along the corridors for one night (in SGD). There will be $2 \leq V \leq 100\,000$ vertices and $1 \leq E \leq 200\,000$ edges in this graph. You have to report two integers as answers: The number of corridors that will be dark that night and the best amount of cost saving that you can do without violating any requirements.

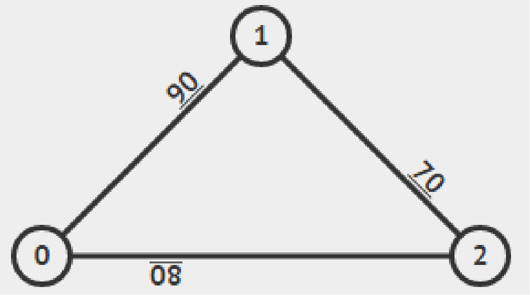

Figure 2: A sample NUS map with 3 buildings and 3 corridors with associated costs.

For example, let's analyze Figure 2 above:

1. Originally, if all lights along three corridors that connect building {0, 1, 2} are turned on, NUS will have to pay electricity bill of 90+70+80=240 SGD that night.

2. If we turn off the lights along corridor that connects building (1-2), then NUS will only have to pay electricity bill of 80+90=170 SGD that night (cost saving of 70 SGD), but **this is not an optimal answer**.

3. If we turn off the lights along two corridors that connect buildings (0-1) and (1-2), then NUS will only have to pay electricity bill of 80 SGD that night ("cost saving" of 160 SGD), but those pulling all-nighters (and security guards) at building 1 that night cannot move safely to building 0 or building 2 as the connecting corridors are all dark, so **this is not valid answer**.

4. **The optimal valid answer** is this: We turn off the lights along corridor that connects building (0-1), then NUS will only have to pay electricity bill of 70+80=150 SGD that night (cost saving of 90 SGD).

Design the best algorithm that you have learned in CS2010 (or beyond) to solve this problem and analyze the time complexity of your algorithm. The rough marking scheme is as follows:

- Not blank, but your algorithm is totally incorrect: 1 mark
- Your algorithm is incorrect: 5 marks
- Your algorithm is correct: 15 marks
- For correct algorithm, we deduct 5 marks for wrong time complexity analysis

Section C.2 Marks = _____

C.3 Avengers Initiative (30 marks)

There are A Avengers ($1 < A \leq 10\,000$, yes, a lot of them now) currently staying in various places (vertices) in the world. Suddenly, there is a major worldwide threat too great for the forces of mankind to handle that requires all of these A Avengers to assemble in one of H S.H.I.E.L.D. Helicarriers ($1 < H \leq 10$, yes, they have more than one now) currently stationed in various places (vertices) in the world. None of the Avenger is currently at any of the Helicarrier at this point of time. So, Nick Fury reactivates the 'Avengers Initiative' and call all these A Avengers to assemble.

Given a world map G that describes up to V places (vertices) in the world ($A+H \leq V \leq 100\,000$) and E weighted edges that describe the fastest direct connection between places in hours ($0 \leq E \leq 200\,000$, edge weight is a positive integer not more than 1000), determine which of the H Helicarriers should be chosen as the meeting point by Nick Fury and compute the shortest possible time (in hours) from Nick Fury's 'Avenger Initiative' reactivation before **all** these A Avengers can meet together and discuss the solution for that worldwide threat. If there is no Helicarrier where all A Avengers can assemble, you have to report 'Doomsday' instead.

For example, you are given a world map G as shown in Figure 3. In this example, the best meeting point is the Helicarrier at vertex 0 and the meeting can begin earliest 6 hours from now as Avenger at vertex 5 is the slowest to arrive (he/she needs 5+1 = 6 hours to traverse path 5 → 1 → 0). The other 3 Avengers will be able to reach Helicarrier at vertex 0 in ≤ 6 hours. Note that if Nick Fury chooses Helicarrier 1 as the meeting point, then Avenger at vertex 4 will cause the bottleneck as he/she needs 4+9 = 13 hours to traverse path 4 → 0 → 1. The world probably have ended by then.

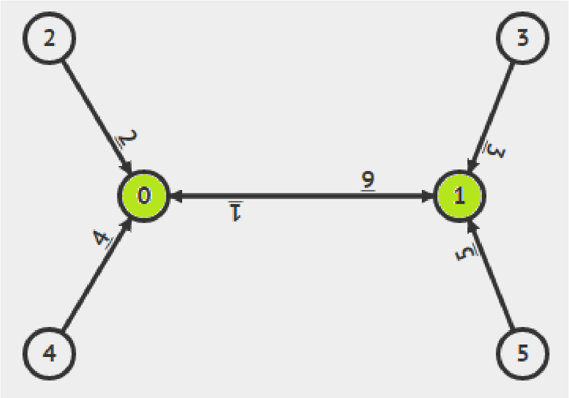

Figure 3: A Sample World Map G. There are $H = 2$ Helicarriers currently located at vertex {0, 1}. There are $A = 4$ Avengers currently located at vertex {2, 3, 4, 5}. For simplicity, no other intermediate vertices are shown in this Sample World Map. Storyline credit: Marvel Comics.

Design the best algorithm that you have learned in CS2010 (or beyond) to solve this problem and analyze the time complexity of your algorithm. The rough marking scheme is as follows:

- Not blank, but your algorithm is totally incorrect: 1 mark
- Your algorithm is incorrect: 5 marks
- Your algorithm is correct, but (much) slower than the examiner's best answer: 15 marks
- Your algorithm is correct and equal to/better than the examiner's best answer: 30 marks
- For correct algorithm, we deduct 5 marks for wrong time complexity analysis

– End of this Paper, All the Best –

Section C.3 Marks = _____

Section C Marks = _____