National University of Singapore

School of Computing

Semester 1 (2015/2016)

CS2010 - Data Structures and Algorithms II

Written Quiz 2 (15%)

Saturday, October 31, 2015, 10.00am-11.30am (90 minutes)

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this assessment paper until you are told to do so.
2. Written Quiz 2 is conducted at COM1-2-SR1 (for all 178 students).
3. This assessment paper contains THREE (3) sections.
   It comprises TEN (10) printed pages, including this page.
4. This is an **Open Book Assessment**.
5. Answer **ALL** questions within the **boxed space** in this booklet.
   You can use either pen or pencil. Just make sure that you write **legibly**!
6. 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.
7. 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.
8. All the best :).
   After Written Quiz 2, this paper will be collected and graded manually as fast as humanly possible. We will try to return the script to you on Wednesday, 04 November 2015 (Lecture 12). If that is not possible, we will do so by Wednesday, 11 November 2015 (the last Lecture 13).
9. Write your Student Number in the box below:

A	0							

A Analysis (15 marks); Marks = _____

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

1. We can design an $O(1)$ algorithm to detect if a **connected undirected graph** with V ($V \geq 3$) vertices and E edges contains a simple cycle involving 3 or more vertices.

2. This time, the graph is **not necessarily connected**. We can design an $O(V)$ algorithm, independent of E (the number of edges), to detect if an **undirected graph** with V ($V \geq 3$) vertices and E edges contains a simple cycle involving 3 or more vertices.

3. The edge weights of a connected undirected weighted graph G are all different. That means the MST of G may be not unique, i.e. there can be MST_1 of G that is structurally different than MST_2 of G yet both MST_1 and MST_2 have the same minimum weight.

4. Suppose that we only run the Bellman Ford's algorithm for one round, i.e. we only relax all E edges once. Such Bellman Ford's variant surely produces wrong answer for *all kind* of graphs.

5. There is only one possible shortest path from source vertex s to a certain destination vertex t in a Directed Acyclic Graph (DAG).

Section A Marks = _____

B Not Yet in VisuAlgo Online Quiz (50 marks)

In this section, Steven design a few questions that *may* become part of VisuAlgo Online Quiz system *in the future*.

B.1 Non Standard DFS Challenge (10 marks)

Suppose that we have a DFS implementation `dfs-avoid-odd` as follows:

```
dfs-avoid-odd(u)
  visited[u] <- 1
  if out-degree(u) is odd, return // additional new line
  for all v adjacent to u // ordered based on increasing vertex number
    if visited[v] = 0
      dfs-avoid-odd(v)
```

In-exam clarification: `out-degree(u)` is the number of neighbors of that vertex u.

Show the sequence of vertex visitation (5 marks) and state which vertices are not reachable (another 5 marks) when we execute `dfs-avoid-odd(0)` and `dfs-avoid-odd(7)` on the graph in Figure 1.

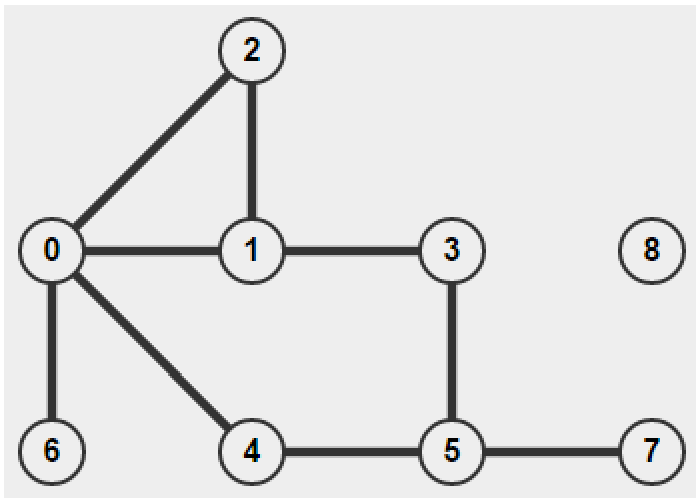

Figure 1: Run `dfs-avoid-odd(0)` and `dfs-avoid-odd(7)` on this graph

`dfs-avoid-odd(0)` will visit (in this order): ______________________________

`dfs-avoid-odd(0)` will not visit: ______________________________

`dfs-avoid-odd(7)` will visit (in this order): ______________________________

`dfs-avoid-odd(7)` will not visit: ______________________________

Page 3 Marks = _____

B.2 Gotta Find 'Em All Challenge (10 marks)

The graph in Figure 2 is a connected unweighted graph with $V = 5$ vertices, i.e. all edge weights are ones. Obviously the MST weight of this graph is 4 units, but that is not the question. The question is how many different MST structures exist in Figure 2?

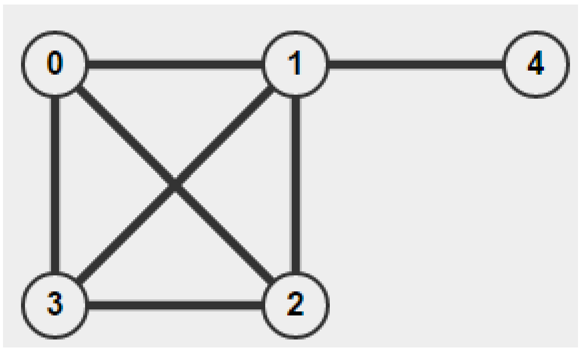

Figure 2: How many different MSTs exist in this picture?

Very short answer: ______________________

In case you over/under count, you may want to draw such different MSTs for 3 partial marks in the space below so that we can give you partial marks for effort. However, if you are very confident that your answer is correct, just state a single correct number and we will give you 10 marks.

Page 3+4 Marks = _____

B.3 Virus Infection Challenge (10 marks)

You define vertices that have shortest path weight $-\infty$ from source vertex 0 as 'virus infected'.

Given the directed weighted graph in Figure 3 that **may have negative weight edge(s) and may have negative weight cycle(s)**, please clearly shade all vertices that are infected by virus.

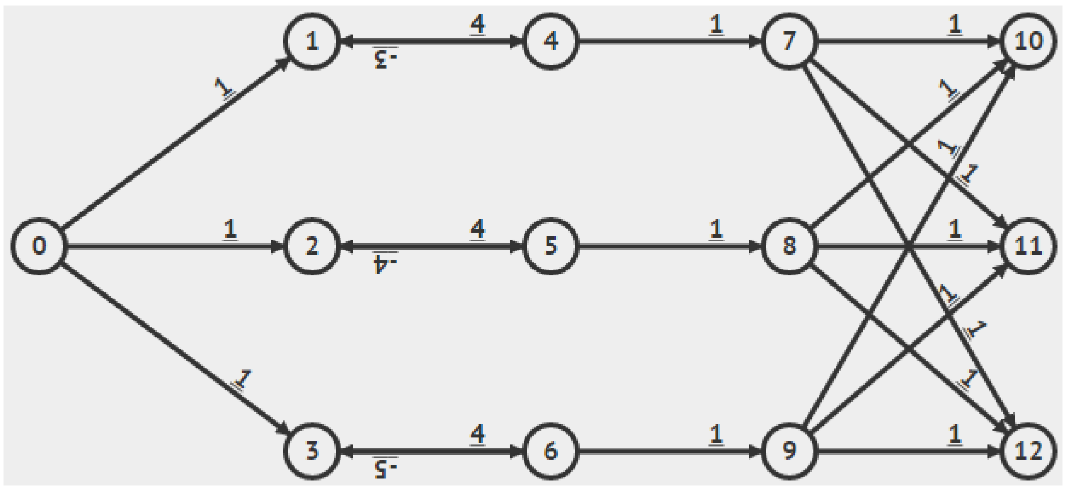

Figure 3: Please clearly shade all vertices that are infected directly in this picture

Please write a **short** explanation on why you shade those vertices:

Page 3+4+5 Marks = _____

B.4 Save Face Challenge (10 marks)

You mistakenly implement BFS algorithm instead of Dijkstra's algorithm to solve an SSSP problem for a graph with $V = 7$ vertices (the vertices are labeled from [0..6]) and $E = 11$ directed weighted edges (with 11 different integer edge weights $\in$ [1..11]). The source vertex is vertex 0. We have placed the vertices on Figure 4. Your job is to draw 11 directed weighted edges on this picture as per instruction so that your BFS algorithm still happen to produce correct answer on the graph that you drawn and thus safe your own face (for now).

In-exam clarification: Source vertex 0 has to be able to reach vertices [1..6].

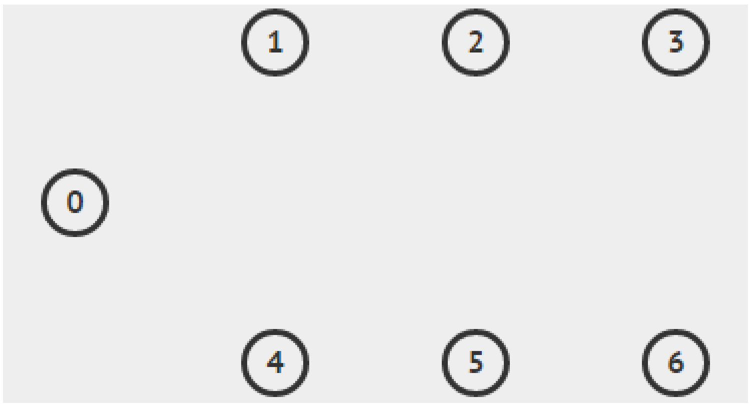

Figure 4: Draw your answer directly in this picture

To help us in grading, **briefly** explains on why BFS algorithm happens to be correct in your drawing:

Page 3+4+5+6 Marks = _____

B.5 Save Face Challenge, Extreme? (10 marks)

After solving the previous Save Face Challenge in Question B.4, you think further...

You have learned about DFS, BFS, Prim's, and Dijkstra's algorithm in the second part of CS2010. You were told that all four algorithms requires a source vertex to begin with and all four algorithms terminate with their own spanning trees of the graph.

Instead of just comparing BFS and Dijkstra's as in Question B.4, you also want to include DFS and Prim's (an MST algorithm). You now want to challenge yourself to construct one single undirected weighted (each edge is treated as having bidirectional edges of same weight by Dijkstra's algorithm) graph with $V = 7$ vertices (the number of edges and the choice of edge weights is now entirely up to you) so that `dfs(0)`, `bfs(0)`, `prim(0)`, and `dijkstra(0)` somehow happen to produce exactly the same spanning tree.

In-exam clarification: The graph that you draw has to be connected.

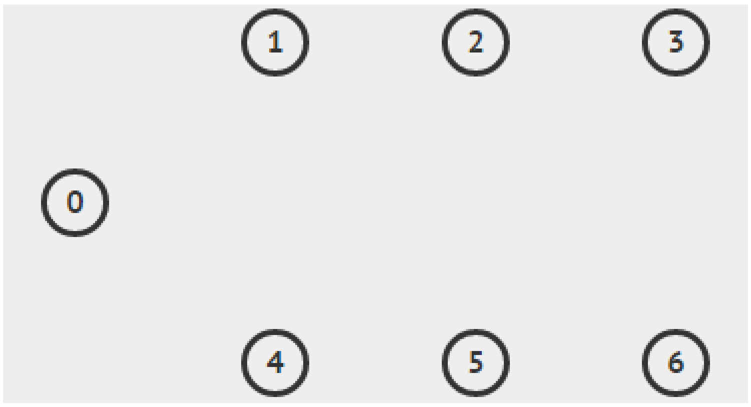

Figure 5: Draw your answer directly in this picture

To help us in grading, **briefly** explains your answer:

Section B Marks = _____

C Think Outside the Box (35 marks)

C.1 Graph Traversal (10 marks)

In Written Quiz 1, you were asked to store graphs with characteristics like the example shown in Figure 6 below: Near complete graphs of V vertices ($5 < V < 100\,000$) with only up to k edges missing ($0 \leq k \leq 7$). To further clarify the question, the graphs are unweighted, the vertices are numbered from [0..V-1], and no special information is stored in each vertex other than vertex number.

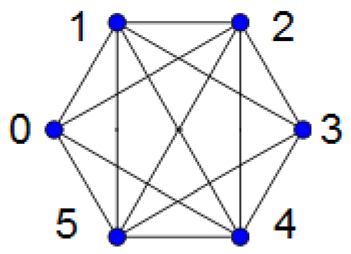

Figure 6: Near Complete Graph with $V = 6$ Vertices (One Edge (0, 3) is Missing)

Now in this Written Quiz 2, you were asked to check if two different vertices i and j are connected in such graph above. What is your best algorithm and analyze it's time complexity?

In-exam clarification: Vertices i and j are connected means that there is a path from i to j, they do not have to be necessarily adjacent.

Page 8 Marks = _____

C.2 Connected Components (25 marks)

You are given a connected undirected unweighted graph G with V vertices and E undirected edges ($1 \le V, E \le 200\,000$). If we delete a vertex, like in PS3, the number of Connected Components (CCs) in G may change. You are given K vertices that you will delete one after another from G. All these K vertices to be deleted are given to you upfront. Now your job is to report the number of CCs in G every time a vertex is deleted from G.

For example, if you are given the connected undirected graph as shown below, deleting $K = 3$ vertices {1, 2, 7} in that order produces 4 CCs ({0}, {4, 5}, {2}, {3, 6, 7, 8}), 3 CCs ({0}, {4, 5}, {3, 6, 7, 8}), 4 CCs again ({0}, {4, 5}, {3, 6}, {8}), respectively.

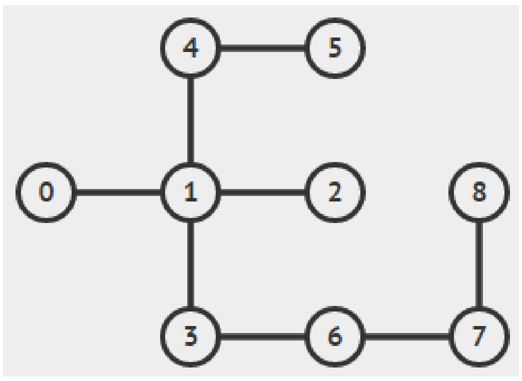

Figure 7: An Example Graph G

Post-exam clarification (for future students who are using this paper for practice): Figure 7 is an example graph G that happens to be a tree. This question is asked on a general graph G. Moreover, students are reminded that the queries of number of CCs are done right after deletion of a vertex but all K vertices to-be-deleted are given to you upfront.

C.2.1 $K = 1$ (10 marks)

How to compute the number of CCs in G if there is only $K = 1$ deleted vertex only?
What is the time complexity of such algorithm?

Page 8+9 Marks = _____

C.2.2 $K \leq V$ (15 marks)

How to compute the number of CCs in G if there can be up to $K \leq V$ deleted vertices? Note that $1 \leq V, E \leq 200\,000$ and any computation above $100M$ steps is considered slow. To get full marks, you need to state your algorithm, analyze it's time complexity, and conclude that it runs in $O(V + E)$.

– End of this Paper, All the Best –

Candidates, please do not touch this table!

Section	Maximum Marks	Your Marks	Comments from Grader
A	15		
B	50		
C	35		
Total	100		