National University of Singapore

School of Computing

Semester 1 (2014/2015)

CS2010 - Data Structures and Algorithms II

Written Quiz 2 (10%)

Saturday, October 25, 2014, 10.00-11.17am (77 mins)

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this question paper until you are told to do so.
2. Written Quiz 2 is conducted at COM1-2-206/SR1.
3. This question paper contains THREE (3) sections with sub-questions.
   It comprises TEN (10) printed pages, including this page.
4. Write all your answers in this question paper, **but only in the space provided**.
   You can use either pen or pencil. Just make sure that you write **legibly**!
   Important tips: Pace yourself! Do **not** spend too much time on one (hard) question.
5. This is an **Open Book Examination**. You can check the lecture notes, tutorial files, problem set files, Steven's 'Competitive Programming' book, or any other books/printed material that you think will be useful. But remember that the more time that you spend flipping through your files implies that you have less time to actually answer the questions.
6. When this Written Quiz 2 starts, **please immediately write your Matric Number here:** ______________________ **(include the last letter) and your Tutorial Group Number/ Tutorial TA Name/Tutorial Day (Tue or Wed)/Tutorial Time:** __________
   do not write your name in order to facilitate unbiased grading.
7. All the best :).
   After Written Quiz 2, this question paper will be collected, graded manually in about 1.5 weeks, and likely returned to you via your Tutorial TA on Week12.

–This page is intentionally left blank. You can use it as ‘rough paper’–

1 Test understanding of BFS/DFS/MST/SSSP (15 Marks)

Grading scheme: 0 (wrong answer), 1 (correct answer).

1. What is the resultant graph after performing a BFS (from any source) called ? ______________
2. In BFS, the vertices to be en-queued after processing the source vertex are all those that ______________ away from the source vertex.
3. If a graph has all edges of weight 3, the best algorithm to find the shortest path from a vertex A to all other vertices is ______________.
4. The minimum number of times Bellman Ford's algorithm need to relax all edges of a graph in order to find the shortest path is ______________.
5. The worst case time complexity of Original Dijkstra's algorithm when run on a tree is ______________.
6. For a very sparse DAG, the best SSSP algorithm to use is ______________.
7. The best SSSP algorithm to use on a graph with -ve edge weights and (and potentially contain -ve cycles) is ______________.
8. A tree is always rooted. (true/false) (circle the answer)
9. There is only 1 unique path between any 2 vertices in a tree. (true/false) (circle the answer)
10. If a new edge connecting 2 existing nodes in a tree is added, a cycle will always be formed. (true/false) (circle the answer)
11. Given the shortest path between 2 vertices A and B, the subpaths of that shortest path are themselves shortest paths. (true/false) (circle the answer)
12. The set of edges in the SSSP spanning tree and the MST for any undirected connected graph with positive edge weights will never be the same. (true/false) (circle the answer)
13. Adding a constant to every edge weight in a graph does not change the solution to the SSSP problem. (true/false) (circle the answer)
14. The largest edge in each cycle of an undirected connected graph with unique edge weights must not be in any valid MST of the graph. (true/false) (circle the answer)
15. The smallest edge in each cycle of an undirected connected graph with unique edge weights must be in every valid MST of the graph. (true/false) (circle the answer)

2 Analysis (15 marks)

Prove (the statement is correct) or disprove (the statement is wrong) the following statements below. 3 marks per each statement below (1 mark for saying correct/wrong, 2 marks for explanation):
Note : You are only given a small amount of space below (i.e. do **not** write too long-winded answer)!

1. For any undirected connected graph where number of edges $E >$ number of vertices V, if there are some edges with the same weight, the MST of the graph will always not be unique. (3 marks)

2. If the weights of all edges in a weighted graph are unique, there is always a unique shortest path from a source to destination in such a graph. (3 marks)

3. The following algorithm to find the largest edge from a given source s to a given destination d in the MST of a graph G is correct.
   Run Kruskal's on G, and keep track of the largest edge $e*$ included in the MST so far. Every time it includes an edge e to be in the MST, we query the UFDS whether s and d are in the same set. If they are, stop Kruskal's and simply output $e*$ as the answer. (3 marks)

4. For modified Dijkstra's, a vertex v cannot be en-queued in the priority queue more times than the number of in-edges to v. (3 marks)

5. In a tree of ≥ 3 vertices, for any three vertices a,b,c, let $E*$ be the set of edges formed from the union of the set of edges in the path $a \sim> b$ and the path $a \sim> c$. The set of edges in the path $b \sim> c$ must then be a subset of the edges in $E*$. (3 marks)

3 Application Questions (20 + 10 Marks)

Please write pseudo-code (which can be english descriptions) for all your algorithms. There should not be any black-boxes in your pseudo-code. This means that if you use any function that is not defined in the lecture notes or a well recognized algorithm, then you must define them.

Q1. SSSP Application: (8 marks)

a.) **(4 marks)** Two friends lives in cities A and B respectively, and they want to meet up. In order to convenience the both of them, they come up with a strategy to meet up at a city C, where C satisfies the following 2 conditions.
Firstly city C satisfies the condition that the sum of their total travelled distance to C is the smallest. Secondly city C satisfies the condition that the absolute difference between the distance travelled by each person is minimized. The first condition is more important than the second.
Come up with an efficient algorithm to find the city they should meet up in. If there are multiple candidate cities which satisfy the above criteria just output one of them. You can assume that there exist at least 1 such city C and also the roads and cities can be represented as an undirected weighted graph, where the edge weight is the distance between the connected cities.

b.) **(4 marks)** Given an undirected graph where edge weights w are multiples of 2 and have values between 2 and 50, find the shortest path from a given source vertex s to all possible destination vertex d using an efficient algorithm.

Q2. MST Application: (12+10 marks)

For some applications of MST, the original graph $G(V, E)$ is not static but dynamic in the sense that there can be changes to its edges and vertices. You may assume for each of the question below that the edge list, adjacency list and current MST is given to you.

a.) **(1 mark)** Give an efficient algorithm to update the MST when a new vertex s of degree 1 is inserted into $G(V, E)$.

b.) **(3 marks)** Give an efficient algorithm to update the MST when a new vertex s of degree 2 is inserted into $G(V, E)$.

c.) **(8 marks)** Give an efficient algorithm to update the MST when an edge (a,b) in $G(V,E)$ changes its weight (can be a decrease or increase).

d.) **(BONUS QUESTION worth 10 marks***)** Give an efficient algorithm that has **worst case time complexity better than** $O(E+VlogV)$ to update the MST when a new vertex s of degree up to $|V-1|$ is inserted into $G(V,E)$. (*This is a bonus question, do not attempt until you have finished all the other questions! Also no marks are given for algorithms with efficiency worse or equal to* $O(E+VlogV)$)

– End of this Paper –

Candidates, please do not touch this table!

Section	Maximum Marks	Student's Marks
1	15	
2	15	
3	min(8+12, your marks)	
Total	50	