CSCI 3104 PS4b

Jonathan Phouminh

TOTAL POINTS

34 / 45

QUESTION 1

10 pts

1.1 3 / 5

✓ **+ 1 pts Maintained separate sets of explored and unexplored vertices**

+ **2 pts** Minimization and picking of the next vertex from the unexplored set and adding it to explored set

✓ **+ 1 pts Partially correct way of picking next vertex from the unexplored set and adding it to explored or explanation not clear. Please refer to the solution file.**

+ **2 pts** Updating the max-weight values of unexplored vertices after a new vertex is added to the explored set.

✓ **+ 1 pts Partially correct update of max-weight values. This is applicable if there is a minor mistake while maintaining the max weight values. Please refer to the solution file.**

+ **0 pts** Incorrect or not attempted

1.2 3 / 5

✓ **+ 1 pts Identified the loop invariant or understood the fact that we have solved the problem for the explored vertices.**

+ **2 pts** Identified the fact that when a new vertex is added to explored it is chosen such that there is no max-weight path to the selected node from a vertex that does not belong to explored set

+ **2 pts** Proof by contradiction or any other strategy

✓ **+ 1 pts Proof by contradiction or any other strategy partially correct**

✓ **+ 1 pts Partially correct Identified the method of picking the right vertex from the unexplored set.**

+ **0 pts** Incorrect or not attempted

QUESTION 2

11 pts

2.1 4 / 4

✓ **+ 4 pts Correct order and correct set of edges: AE -> EF -> FB/EB -> FG -> GH -> GC -> GD**

+ **3 pts** Slightly incorrect set of edges/order

+ **2 pts** Partially correct set of edges/partially correct order

+ **1 pts** Incorrect set of edges or incorrect correct order

+ **0 pts** Incorrect

2.2 7 / 7

✓ **+ 2 pts Correct set of edges: AE -> EF -> FB/EB -> FG -> GH -> GC -> GD or other alternate solution**

✓ **+ 2 pts Correct order of edges**

✓ **+ 3 pts Correct Partitions: [{A,B,C,D}, {E,F,G,H}], [{A,B,C,D,E}, {F,G,H}], [{A,E,F,G,H}, {B,C,D}], [{A,B,E}, {C,D,F,G,H}], [{A,B,E,F,G}, {C,D,H}], [{A,B,E,F,G,H}, {C,D}], [{A,B,C,E,F,G,H}, {D}] or other alternate solution**

+ **1.5 pts** Mostly correct set of edges

+ **1.5 pts** Mostly correct order

+ **2.5 pts** Mostly correct partitions

+ **1 pts** Partialy correct set of edges

+ **1 pts** Partially correct order

+ **1.5 pts** Partially correct partitions

+ **0 pts** Incorrect

QUESTION 3

3 10 / 10

✓ **+ 2 pts Got the answer right (case 1: If reducing from T then it should not change, case 2: otherwise it could change)**

+ **8 pts** case 2 : gave a great counterexample or a reasonable explanation.

+ **4 pts** case 1 : No logical errors in the proof.

+ **2 pts** case 1 : gave a reasonable premise

+ **2 pts** case 1 : the reason why T will still be a MST is clear.

+ **0 pts** no solution / wrong

✓ + **8 pts** **Got the idea right but need a more rigorous solution next time. (Please see the answer and try to imitate that)**

QUESTION 4

14 pts

4.1 2 / 7

+ **3 pts** Get conclusion correctly. (No)

+ **4 pts** Give counterexample correctly.

✓ + **2 pts** **Know the difference between MLST and MST, but failed to give a counterexample.**

+ **0 pts** Incorrect/Not attempted.

Very good analysis between MLST and MST, but why the rest of the edges are also minimized?

4.2 5 / 7

✓ + **2 pts** **Identified the contradiction correctly**

+ **5 pts** Correct proof

+ **3 pts** Proof is not explained clearly

✓ + **3 pts** **Knows the difference between MST and MLST but failed to provide correct proof**

+ **0 pts** Incorrect answer or not attempted

+ **4 pts** Partially correct proof

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**

Problem Set 4b (45 points) **Fall 2019, CU-Boulder**

Instructions for submitting your solution:

- The solutions **should be typed** and we cannot accept hand-written solutions. Here's a short intro to Latex.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Gradescope if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.
- Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.
- For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.
- You may work with other students. However, **all solutions must be written independently and in your own words.** Referencing solutions of any sort is strictly prohibited. You must explicitly cite any sources, as well as any collaborators.

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 4b (45 points) **Fall 2019, CU-Boulder**

1. (10 pts) For a directed graph with positive weights, we define the max-weight of a path from s to d as the maximum of the edge weights along the path. For example, if the path from A to D has edges and weights of $e_{AB} = 5$, $e_{BC} = 4$, and $e_{CD} = 1$, the length of the path is defined as $e_{AB} + e_{BC} + e_{CD}$, and the max-weight is 5.

   (a) (5 pts) Give an algorithm to compute the smallest max-weight paths from a source vertex s to all other vertices. In this problem, you are changing the definition of length of the path from A to D to $max(e_{AB}, e_{BC}, e_{CD})$ (Hint: Your algorithm should be a modification of Dijkstra's algorithm presented in Lecture.)

   Solution.

   Algorithm (Psuedo-code):

   While X $\neq V$:

   $-$among all edges$(v, w)\in$ with E with V $\in$ X, W $\notin$ X, pick one that minimizes.

   -For the first iteration put that edge as the current edge with the highest weight, replace after every iteration if the next edge's weight is greater than the previous,save this in variable called, MaxEdge.

   - A[v] + l_{vw}

   $-add \;\; W \;\; to \;\; X$

   $-setA[W^*] = A[V^*] + l^{V*W*}$

   $-setB[W^*] = B[V^*]U(V^*, W^*)$

   $Once \;\; while \;\; loop \;\; ends, \;\; return \;\; MaxEdge.$

Name: Jonathan Phouminh
ID: 106054641

(b) (5 pts) Prove the correctness of your algorithm.

Solution.
We will prove this by loop invariant.
LoopInvariant **:** At the beginning of the i^{th} iteration, MaxEdge, has the max edge weight for the current shortest path from vertex S - V.

Initialization **:** During the first execution of the loop our modification of Dijkstra's algorithm will have only selected / seen one edge therefore it has to be the maximum weighted edge from the path from S - V.

Maintenance **:** At the i^{th} iteration of the loop there will be two cases.
i) The edge that was chosen next by Dijkstra's does not have a weight greater than the edge in MaxEdge, thus, MaxEdge will remain the same.
ii) The edge chosen does have a weight greater than the current edge in MaxEdge and MaxEdge will be updated to the edge that was just selected by Dijkstra's in the current iteration.

Termination **:** Our loop will terminate when X = V and the max weight of the shortest path will be returned. This is easy to verify, since Dijkstra's gives us the shortest path, we can trace it and verify that the heaviest edge was returned. QED

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 4b (45 points) **Fall 2019, CU-Boulder**

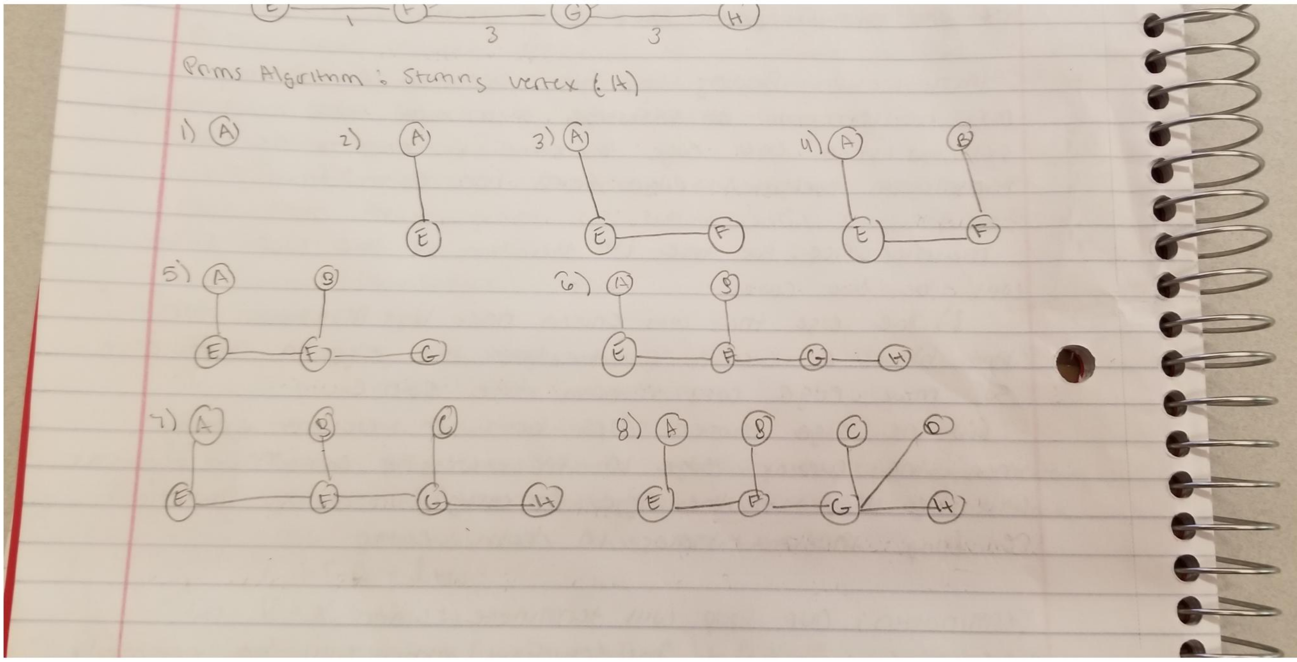
2. (11 pts) Based on the following graph :

`PS4/mst_graph_q2.jpg`

(a) (4 pts) In what order would Prim's algorithm add edges to the MST if we start at vertex A?

Solution.

Prims would add them in this order: A, E, F, B, G, H, C, D



Name: Jonathan Phouminh
ID: 106054641

(b) (7 pts) In what order Kruskal's would add the edges to the MST? For each edge added by Kruskal's sequentially, give a cut that justifies it's addition.

Solution.

(X,Y) , represents edge between path between X Y
1) (A,E) , satisfied in the cut p_1 ={A, B, C, D} $p_2 =\{E, F, G, H\}$
2) (E,F) , satisfied in the cut p_1 ={A, E} $p_2 =\{B, C, D, F, G, H\}$
3) (F,B) , satisfied in the cut p_1 ={A, E, B} $p_2 =\{F, C, G, D, H\}$
4) (F,G) , satisfied in the cut p_1 ={A, E, B, F} $p_2 =\{C, G, D, H\}$
5) (G,H) , satisfied in the cut p_1 ={A, B, C, E, F, G} $p_2 =\{D, H\}$
6) (G,C) , satisfied in the cut p_1 ={A, B, E, F, G, H} $p_2 =\{C, D\}$
7) (G,D) , satisfied in the cut p_1 ={A, B, C, E, F, G, H} $p_2 =\{D\}$

Order of vertices visited: A, E, F, B, G, H, C, D

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 4b (45 points) **Fall 2019, CU-Boulder**

3. (10 pts) Let T be a MST of a given graph G. Will T still be the MST if we reduce the weight of exactly one of the edges in T by a constant $\mathbf{c}$? Prove your answer.

   Solution.

   Suppose T won't be a minimum spanning tree after reducing on edge by C.

   This implies that there is another spanning tree such that its total weight cost is less than T.

   By definition a minimum spanning tree is a path that touches all vertices with minimum cost. If T was the minimum spanning tree initially then there cannot exist a tree smaller in than itself.

   Therefore, by contradiction if an edge in T is reduced by some constant C, it will still be the MST, just with a smaller total weight cost.

Name: Jonathan Phouminh
ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 4b (45 points) **Fall 2019, CU-Boulder**

4. (14 pts) One of the uses of MSTs is finding a set of edges that span a network for minimum cost. Network problems could also have another type of objective: designing a spanning tree for which the most expensive edge is minimized. Specifically, let $G = (V, E)$ be a connected graph with n vertices, m edges, and positive edge costs that you may assume are all distinct. Let $T = (V, E)$ be a spanning tree of G; we define the **limiting edge** of T to be the edge of T with the greatest cost. A spanning tree T of G is a minimum-limiting spanning tree if there is no spanning tree T of G with a cheaper limiting edge.

   (a) (7 pts) Is every minimum-limiting tree of G an MST of G? Prove or give a counterexample.

   Solution.

   Minimum-Limiting spanning tree: A spanning tree whose limiting edge is minimized.
   Minimum spanning tree: A set of edges that spans a network at a minimum cost.

   By definition of a limiting-minimum tree we know that its total cost is limited by it's heaviest edge. If its heaviest edge is minimized we know that the rest of the edges are also minimized because they can't be bigger than its heaviest edge, therefore forming a cheapest network path.

   Since it is the cheapest path it is also an MST of G.

Name: Jonathan Phouminh
ID: 106054641

(b) (7 pts) Prove that every MST of G is a minimum-limiting tree of G. [**Hint:** Let T be an MST of G, and let T' be a minimum-limiting tree of G. If T is not a minimum-limiting tree, can we replace the heaviest edge of T? Think about how to use T' here.]

Solution.

Assume you have an MST that is not a minimum limiting tree, this implies that we have a spanning tree whose heaviest edge isn't minimized.

This violates the cut property such that the edge taken by the MST was not the minimum edge crossing.

If our MST had an edge that is not minimal, then this implies that the wrong edge was obtained during the heaviest edge's cut.

An MST is a tree whose edges are minimized, therefore by contradiction we cannot have MST whose edge is not minimized. Therefore the heaviest edge of an MST is minimized, thus an MST is an MLT.

Collaborated with: Bao Nguyen, Zachary Chommala

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 4b (45 points) **Fall 2019, CU-Boulder**

Ungraded questions - These questions are for your practice. We won't grade them or provide a typed solution but we are open to discuss these in our OHs and you should take feed backs on your approach during the OHs. These questions are part of the syllabus.

1. Suppose you are given the minimum spanning tree T of a given graph G (with n vertices and m edges) and a new edge $e = (u, v)$ of weight w that will be added to G. Give an efficient algorithm to find the MST of the graph $G \cup e$, and prove its correctness. Your algorithm should run in $O(n)$ time.

   Solution.

2. Based on the following graph :

   `PS4/mst_graph_q2.jpg`

   (a) Run Kruskal's and find the MST. You can break the ties however you want. Draw the MST that you found and also find it's total weight.
   Solution.

(b) Run Prim's starting from vertex A and find the MST. You can break the ties however you want. Draw the MST that you found and also find it's total weight. Is the total weight same as what you get from the above?

Solution.

3. Consider the following unweighted graph, and assign the edge weights (using positive integer weights only), such that the following conditions are true regarding minimum spanning trees (MST) and single-source shortest path (SSSP) trees:

   - The MST is distinct from any of the seven SSSP trees.
   - The order in which Prim's algorithm adds the safe edges is different from the order in which Kruskal's algorithm adds them.

   Justify your solution by (i) giving the edges weights, (ii) showing the corresponding MST and all the SSSP trees, and (iii) giving the order in which edges are added by each of the three algorithms.

   `graph_mst.pdf`

   Solution.