CSCI 3104 Spring 2022
Instructor: Profs. Chen and Layer

Problem Set 3

Due Date .... **February 8, 2022**
Name .... **Your Name**
Student ID .... **Your Student ID**
Collaborators .... **List Your Collaborators Here**

Contents

1 Instructions

- The solutions **must be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

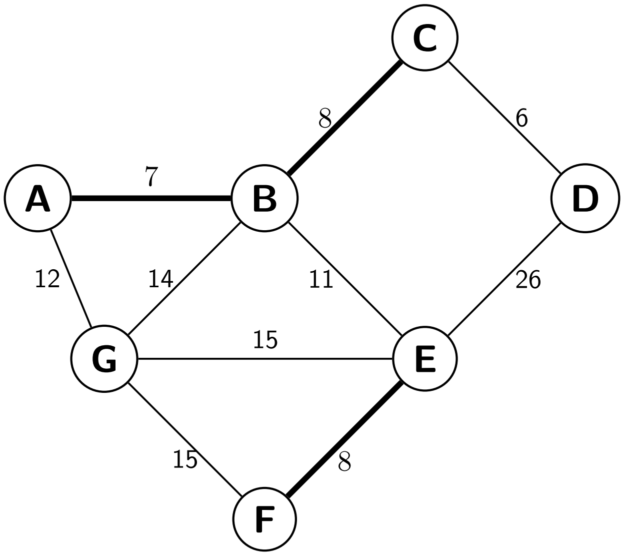
Problem 1.

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

Agreed (signature here). □

3 Standard 7 - MST: safe and useless edges

Problem 2. Consider the weighted graph $G(V, E, w)$ below. Let $\mathcal{F} = \{\{A, B\}, \{B, C\}, \{E, F\}\}$ be an intermediate spanning forest (indicated by the thick edges below). Label each edge that is **not** in $\mathcal{F}$ as safe, useless, or undecided. Provide a 1-2 sentence explanation for each such edge.

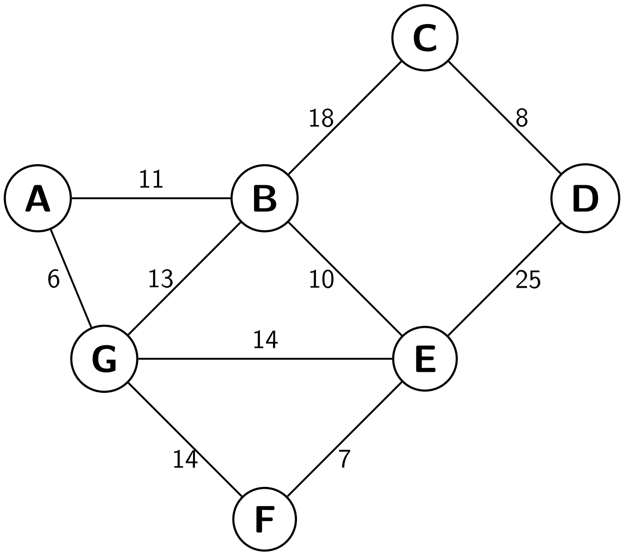


Answer. CD: Safe because it is the next minimum edge so it will be added next.
DE: Useless because it is uneccessary and connects vertices that are already in the graph.
BE: Safe because this is the most efficient way to connect the two parts of the tree.
GE: Useless because it is unecessary and connects vertices that are already in the graph.
BG: Useless because it is unecessary and connects vertices that are already in the graph.
AG: Safe because after CD, it is the next minimum weight edge that would be added and only A is already in F.
GF: Useless because it is unecessary and connects vertices that are already in the graph.

□

4 Standard 8- Kruskal's Algorithm

Problem 3. Consider the weighted graph $G(V, E, w)$ below. Clearly list the order in which Kruskal's algorithm adds edges to a minimum-weight spanning tree for G. Additionally, clearly articulate the steps that Kruskal's algorithm takes as it selects the first **three** edges.

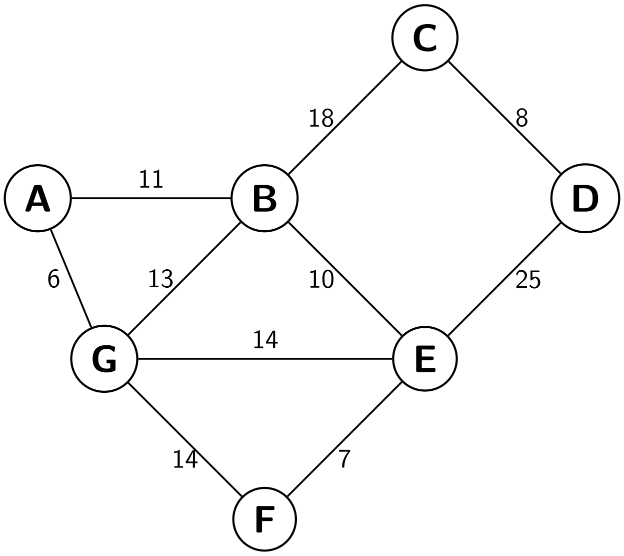


Answer. Edges added in order:
AG (6)
EF (7)
CD (8)
BE (10)
AB (11)
BC (18)

Kruskal's algorithm always adds the minimum cost edge unless it creates a cycle. The edges do not have to be connected to vertices previously added to minimum weight spanning tree. In this case, to start, the algorithm will add AG (with weight 6) because it is the smallest weight. Then, it will add EF (weight 7) because that is the second smallest weight. Third, it will add CD (weight 8), the third smallest weight. □

5 Standard 9- Prim's Algorithm

Problem 4. Consider the weighted graph $G(V, E, w)$ below. Clearly list the order in which Prim's algorithm, **using the source vertex** A, adds edges to a minimum-weight spanning tree for G. Additionally, clearly articulate the steps that Prim's algorithm takes as it selects the first **three** edges.



Answer. Edges added in order:
AG (6)
AB (11)
BE (10)
EF (7)
BC (18)
CD (8)

Starting at vertex A, Prim's algorithm adds the smallest edge connected to A and the corresponding vertex connected by that edge. In this case, from A, we have a choice between AB (weight 11) and AG (weight 6). Prim's algorithm will add AG because it has a smaller weight. Now, the algorithm will compare all the edges A and G are connected to and add the smallest one. In this case, we have the choices of AB (11), GB (13), and GE (14). AB (11) is the smallest, so it will be added next. Now, the algorithm compares the edges connected to vertices A¡ B, and G. In this case, we have the edges GB (13) , GE (14), GF (14), and BE (10). BE (10) will be added third. □