CSCI 3104 Fall 2021
Instructors: Profs. Grochow and Waggoner

Midterm 1- Standard 8

Due Date .... October / $11th$ / 2021
Name .... **Michael Ghattas**
Student ID **109200649**

Contents

1 Instructions

- The solutions **should 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 **may not collaborate with other students**. **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, Discord, 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.

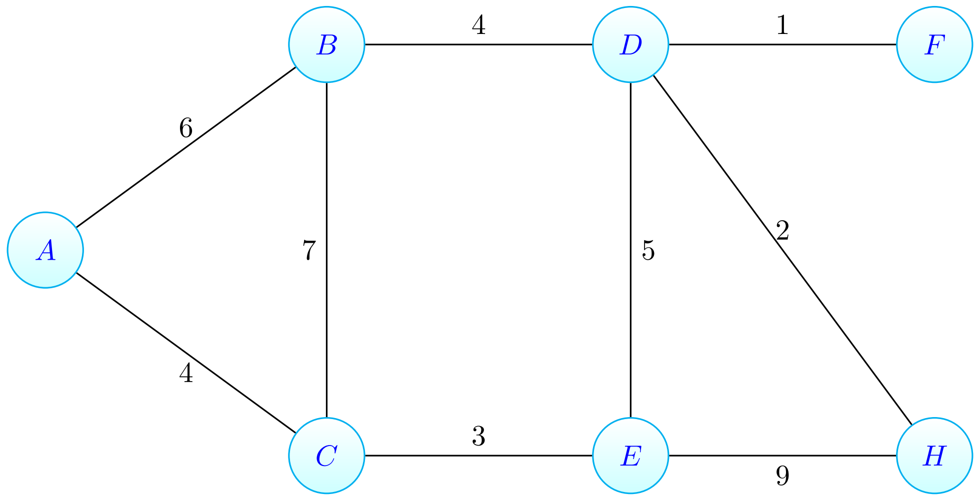
- My submission is in my own words and reflects my understanding of the material.
- I have not collaborated with any other person.
- I have not posted to external services including, but not limited to Chegg, Discord, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

I agree to the above, Michael Ghattas. □

3 Standard 8- Prim's Algorithm

3.1 Problem 2

Problem 2. Consider the following graph $G(V, E, w)$. Clearly indicate the order in which Prim's algorithm adds the edges to the minimum-weight spanning tree **using F as the source vertex**. You may simply list the order of the edges; it is not necessary to exhibit the state of the algorithm at each iteration.



Answer:

We initialize the intermediate spanning forest $\mathcal{F}$ to contain all the vertices of G, but no edges. We then initialize the priority queue to contain the edges incident to our source vertex F.

$Q = [(\{F, D\}, 1)]$

1. We poll the edge $\{F, D\}$ from the queue and mark $\{F, D\}$ as processed. Note that $w(\{F, D\}) = 1$. As $\{F, D\}$ has exactly one endpoint on the component containing F (which is the isolated vertex F), we add $\{F, D\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to D.

$Q = [(\{D, H\}, 2), (\{B, D\}, 4), (\{D, E\}, 5)]$

2. We poll the edge $\{D, H\}$ from the queue and mark $\{D, H\}$ as processed. Note that $w(\{D, H\}) = 2$. As $\{D, H\}$ has exactly one endpoint on the component containing F (which is the isolated vertex $\{F, D\}$), we add $\{D, H\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to H.

$Q = [(\{B, D\}, 4), (\{D, E\}, 5), \{E, H\}, 9)]$

3. We poll the edge $\{B, D\}$ from the queue and mark $\{B, D\}$ as processed. Note that $w(\{B, D\}) = 4$. As $\{B, D\}$ has exactly one endpoint on the component containing F (which is the isolated vertex $\{F, D, H\}$), we add $\{B, D\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to B.

$Q = [(\{D, E\}, 5), (\{A, B\}, 6), (\{B, C\}, 7), \{E, H\}, 9)]$

4. We poll the edge $\{D, E\}$ from the queue and mark $\{D, E\}$ as processed. Note that $w(\{D, E\}) = 5$. As $\{D, E\}$ has exactly one endpoint on the component containing F (which is the isolated vertex $\{F, D, H, B\}$), we add $\{D, E\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to E.

$Q = [(\{C, E\}, 3), (\{A, B\}, 6), (\{B, C\}, 7), \{E, H\}, 9)]$

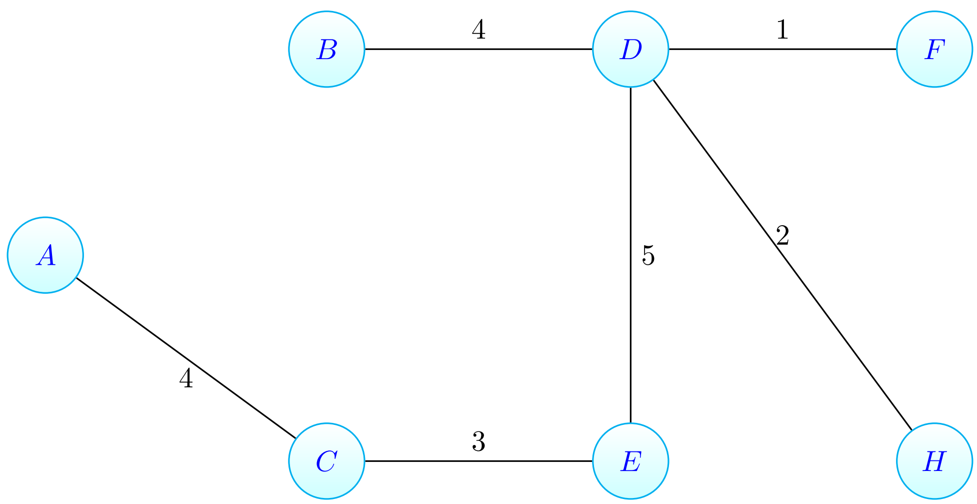
5. We poll the edge $\{C, E\}$ from the queue and mark $\{C, E\}$ as processed. Note that $w(\{C, E\}) = 3$. As $\{C, E\}$ has exactly one endpoint on the component containing F (which is the isolated vertex $\{F, D, H, B, E\}$), we add $\{C, E\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to C.

$Q = [(\{A, C\}, 4), (\{A, B\}, 6), (\{B, C\}, 7), \{E, H\}, 9)]$

6. We poll the edge $\{A, C\}$ from the queue and mark $\{A, C\}$ as processed. Note that $w(\{A, C\}) = 4$. As $\{A, C\}$ has exactly one endpoint on the component containing F (which is the isolated vertex $\{F, D, H, B, E, C\}$), we add $\{A, C\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to A.

$Q = [(\{A, B\}, 6), (\{B, C\}, 7), \{E, H\}, 9)]$

Now there are $[V(G) = 7)]$ *vertices* and $\mathcal{F}$ has $[V(G) - 1 = 6)]$ *edges*. Prim's Algorithm terminates and returns $\mathcal{F}$, which is our minimum-weight spanning tree, illustrated below:



$\square$