CSCI 3104 Fall 2021
Instructor: Profs. Grochow and Waggoner

Problem Set 3

Due Date .... **September 21, 2021**
Name .... **Michael Ghattas**
Student ID **109200649**
Collaborators .... **Me, Myself, and I**

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)

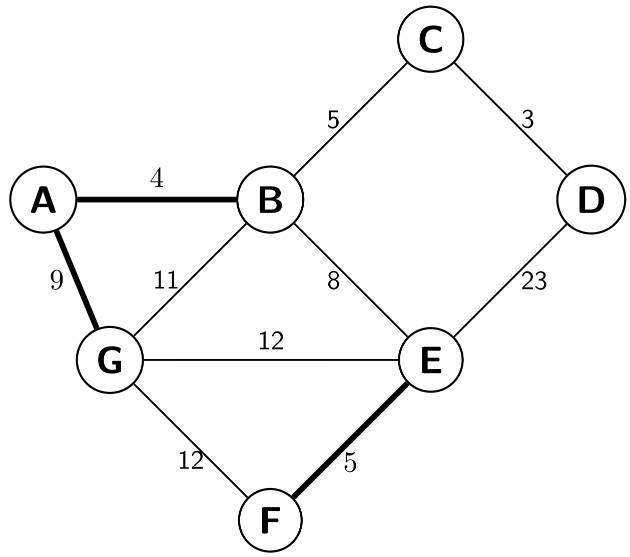
Problem 1.

- 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.

I agree to the above, Michel Ghattas. $\square$

3 Standard 6 - MST: safe and useless edges

Problem 2. Consider the weighted graph $G(V, E, w)$ below. Let $\mathcal{F} = \{\{A, B\}, \{A, G\}, \{E, F\}\}$ 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:

- $\{B, C\} \to$ **Safe:** It is a minimum-weight edge incident to $\{C\}$. Therefore, $\{B, C\}$ is a light edge with exactly one endpoint belonging to $\{C\}$, and the other to $\{G, A.B\}$. Thus $\{B, C\}$ is safe with respect to $\mathcal{F}$.

- $\{C, D\} \to$ **Safe:** It is a minimum-weight edge incident to $\{D\}$. Therefore, $\{C, D\}$ is a light edge with exactly one endpoint belonging to $\{D\}$, and the other to $\{C\}$. Thus $\{C, D\}$ is safe with respect to $\mathcal{F}$.

- $\{D, E\} \to$ **Safe:** It is a minimum-weight edge incident to $\{D\}$. Therefore, $\{D, E\}$ is a light edge with exactly one endpoint belonging to $\{D\}$, and the other to $\{E, F\}$. Thus $\{D, E\}$ is safe with respect to $\mathcal{F}$.

- $\{E, B\} \to$ **Safe:** It is the minimum-weight edge with exactly one endpoint in the component $\{E, F\}$ (as well as being the minimum-weight edge with exactly one endpoint in the component $\{G, A, B\}$). Thus $\{E, B\}$ is safe with respect to $\mathcal{F}$.

- $\{B, G\} \to$ **Useless:** The edge $\{B, G\}$ creates has both endpoints in the component $\{G, A.B\}$. So $\{B, G\}$ is useless with respect to $\mathcal{F}$.

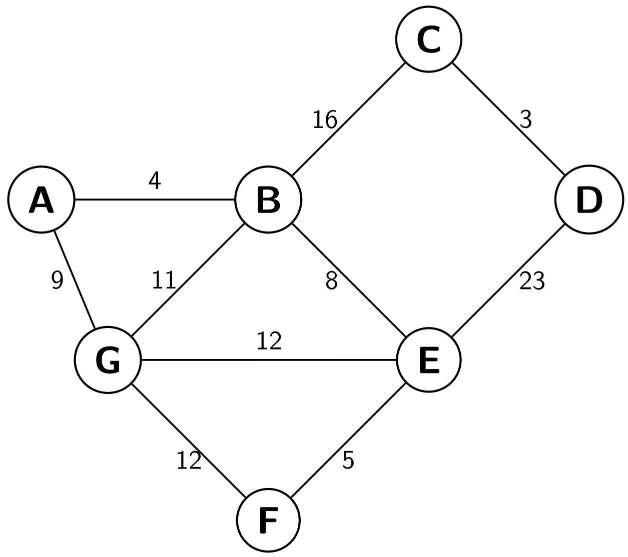
- $\{G, F\} \to$ **Undecided:** While the edge $\{G, F\}$ connects the components $\{G, A, B\}$ and $\{E, F\}$, $\{G, F\}$ is not a minimum-weight edge doing so. Therefore, $\{G, F\}$ is undecided with respect to $\mathcal{F}$.

- $\{G, E\} \to$ **Undecided:** While the edge $\{G, E\}$ connects the components $\{G, A, B\}$ and $\{E, F\}$, $\{G, E\}$ is not a minimum-weight edge doing so. Therefore, $\{G, E\}$ is undecided with respect to $\mathcal{F}$.

$\square$

4 Standard 7- 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.

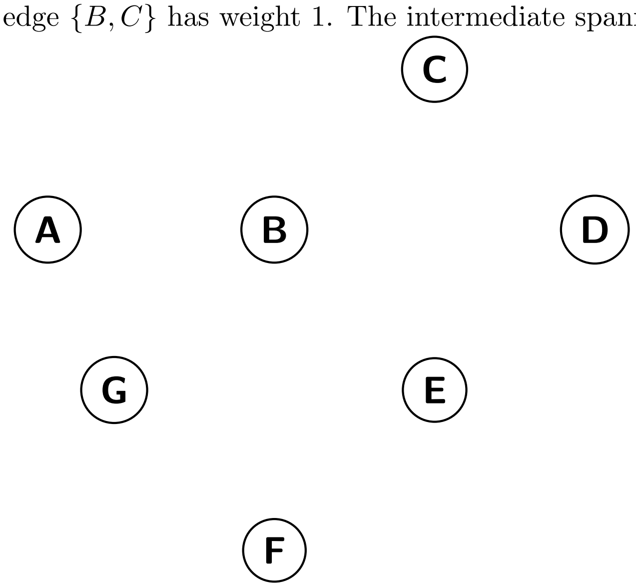


Answer:

1. We initialize the intermediate spanning forest $\mathcal{F}$ to be the empty graph (the graph on no edges). We also place the edges of G into a priority queue, which we call Q.

$$Q = [(\{C, D\}, 3), (\{A, B\}, 4), (\{E, F\}, 5), (\{E, B\}, 8), (\{G, A\}, 9), (\{G, B\}, 11), (\{G, F\}, 12), (\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

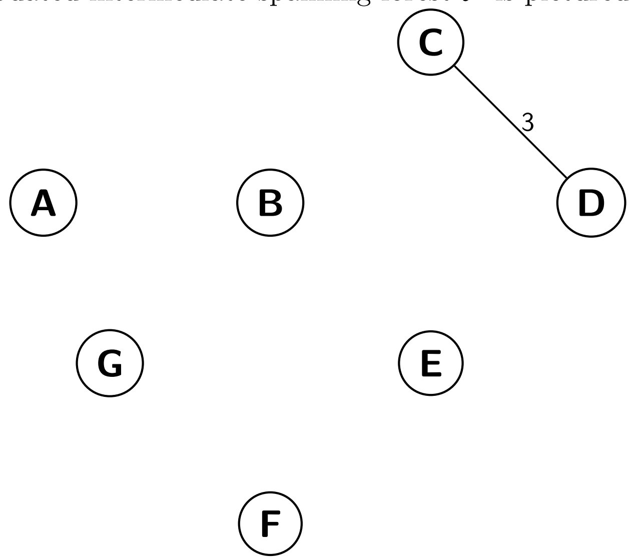
Here, $(\{C, D\}, 3)$ indicates the edge $\{B, C\}$ has weight 1. The intermediate spanning forest $\mathcal{F}$ is pictured below:



2. We poll from Q, which returns the edge $\{C, D\}$. Note that $w(\{C, D\}) = 3$. As C and D are on different components of $\mathcal{F}$, we add the edge $\{C, D\}$ to $\mathcal{F}$.

$$Q = [(\{A, B\}, 4), (\{E, F\}, 5), (\{E, B\}, 8), (\{G, A\}, 9), (\{G, B\}, 11), (\{G, F\}, 12), (\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

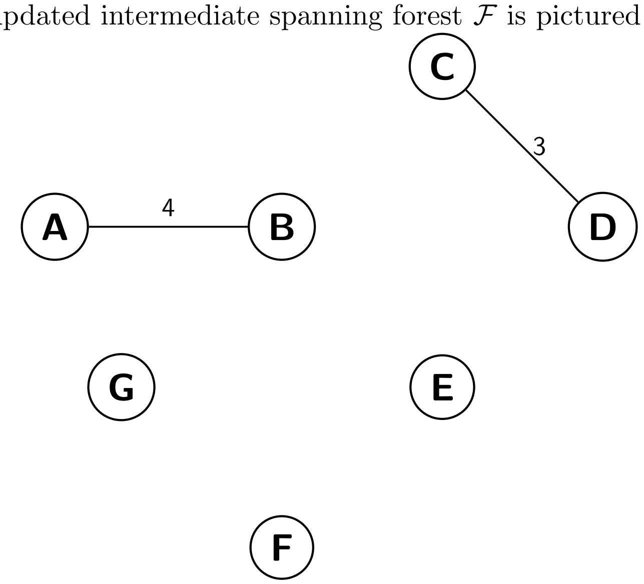
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



3. We poll from Q, which returns the edge $\{A, B\}$. Note that $w(\{A, B\}) = 4$. As A and B are on different components of $\mathcal{F}$, we add the edge $\{A, B\}$ to $\mathcal{F}$.

$$Q = [(\{E, F\}, 5), (\{E, B\}, 8), (\{G, A\}, 9), (\{G, B\}, 11), (\{G, F\}, 12), (\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

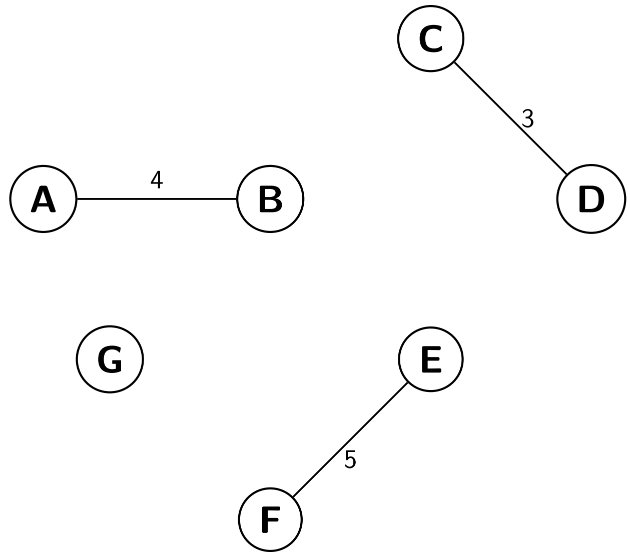
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



4. We poll from Q, which returns the edge $\{E, F\}$. Note that $w(\{E, F\}) = 5$. As E and F are on different components of $\mathcal{F}$, we add the edge $\{E, F\}$ to $\mathcal{F}$.

$$Q = [(\{E, B\}, 8), (\{G, A\}, 9), (\{G, B\}, 11), (\{G, F\}, 12), (\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

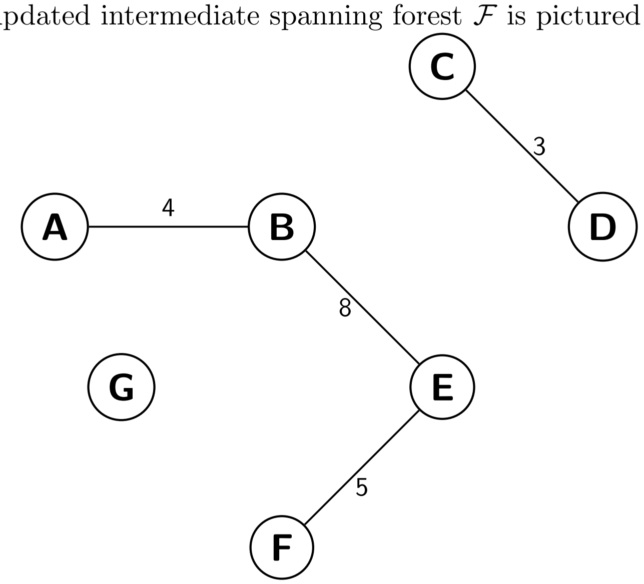
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



5. We poll from Q, which returns the edge $\{E, B\}$. Note that $w(\{E, B\}) = 8$. As E and B are on different components of $\mathcal{F}$, we add the edge $\{E, B\}$ to $\mathcal{F}$.

$$Q = [(\{G, A\}, 9), (\{G, B\}, 11), (\{G, F\}, 12), (\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

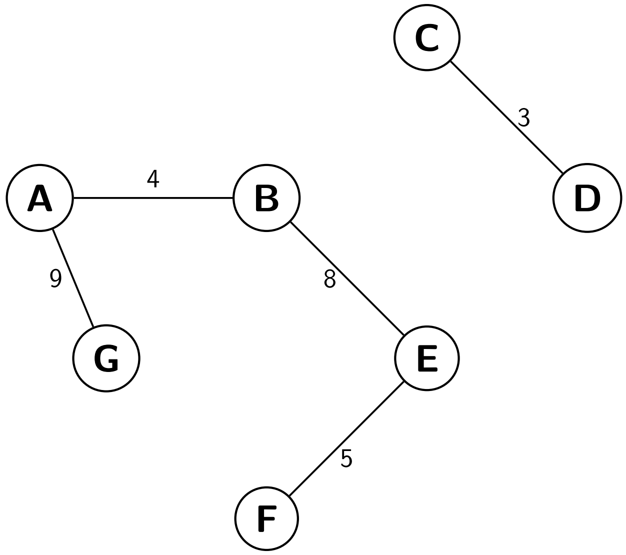
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



6. We poll from Q, which returns the edge $\{G, A\}$. Note that $w(\{G, A\}) = 9$. As G and A are on different components of $\mathcal{F}$, we add the edge $\{G, A\}$ to $\mathcal{F}$.

$$Q = [(\{G, B\}, 11), (\{G, F\}, 12), (\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

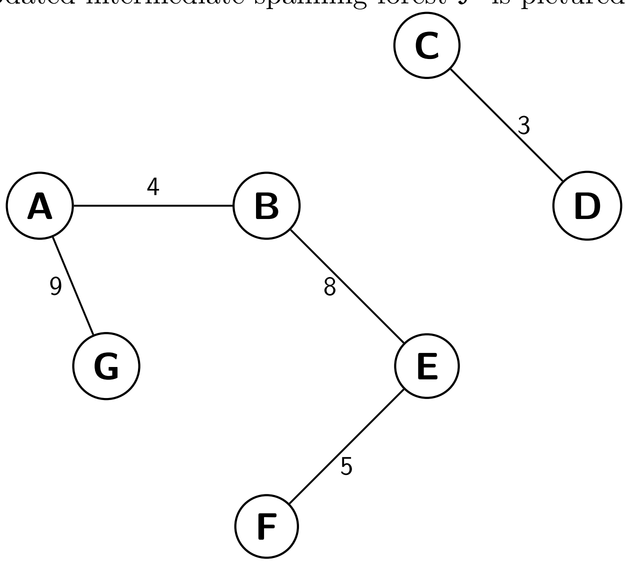
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



7. We poll from Q, which returns the edge $\{G, B\}$. Note that $w(\{G, B\}) = 11$. As G and B are on the same components of $\mathcal{F}$, we do **not** add the edge $\{G, B\}$ to $\mathcal{F}$.

$$Q = [(\{G, F\}, 12), (\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

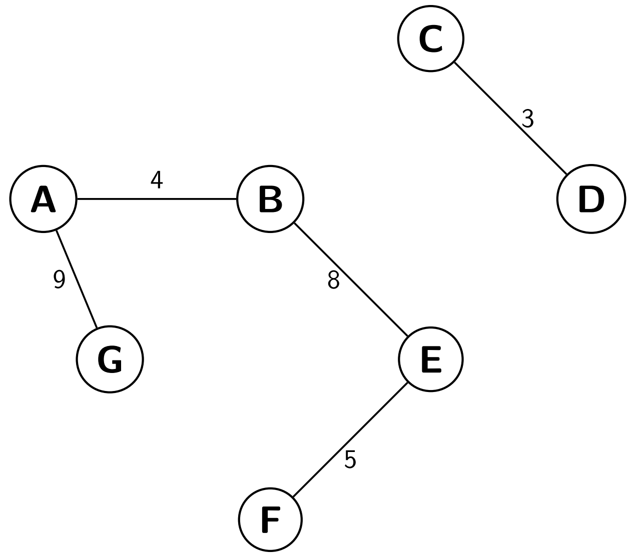
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



8. We poll from Q, which returns the edge $\{G, F\}$. Note that $w(\{G, F\}) = 12$. As G and F are on the same components of $\mathcal{F}$, we do **not** add the edge $\{G, F\}$ to $\mathcal{F}$.

$$Q = [(\{G, E\}, 12), (\{B, C\}, 16), (\{D, E\}, 23)]$$

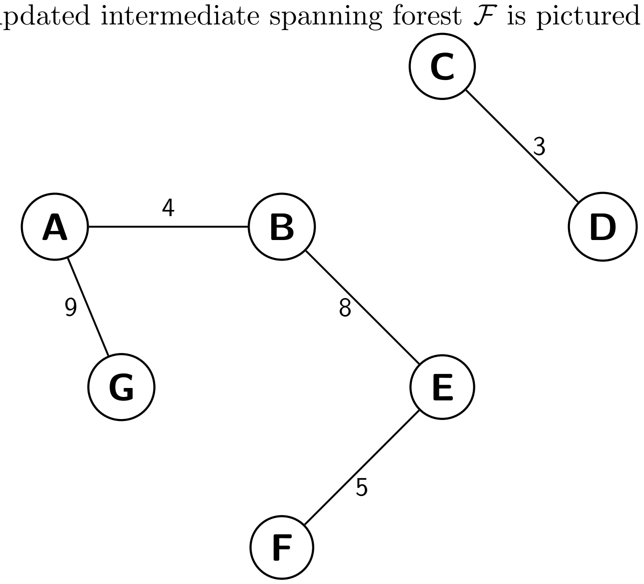
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



9. We poll from Q, which returns the edge $\{G, E\}$. Note that $w(\{G, E\}) = 12$. As G and E are on the same components of $\mathcal{F}$, we do **not** add the edge $\{G, E\}$ to $\mathcal{F}$.

$$Q = [(\{B, C\}, 16), (\{D, E\}, 23)]$$

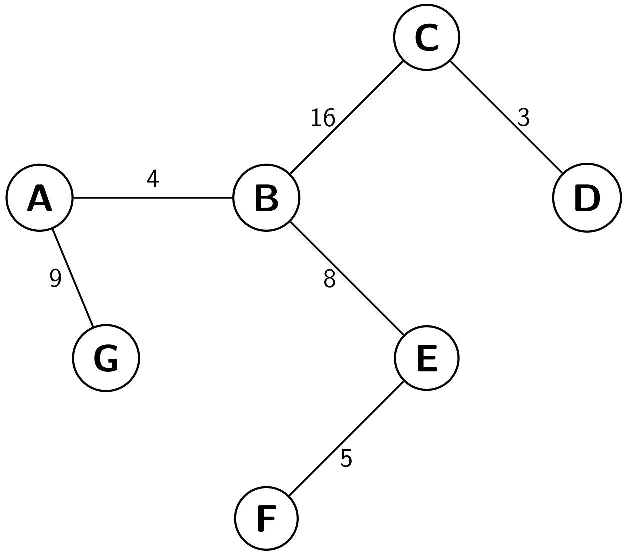
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



10. We poll from Q, which returns the edge $\{B, C\}$. Note that $w(\{B, C\}) = 16$. As B and C are on different components of $\mathcal{F}$, we add the edge $\{B, C\}$ to $\mathcal{F}$.

$$Q = [(\{D, E\}, 23)]$$

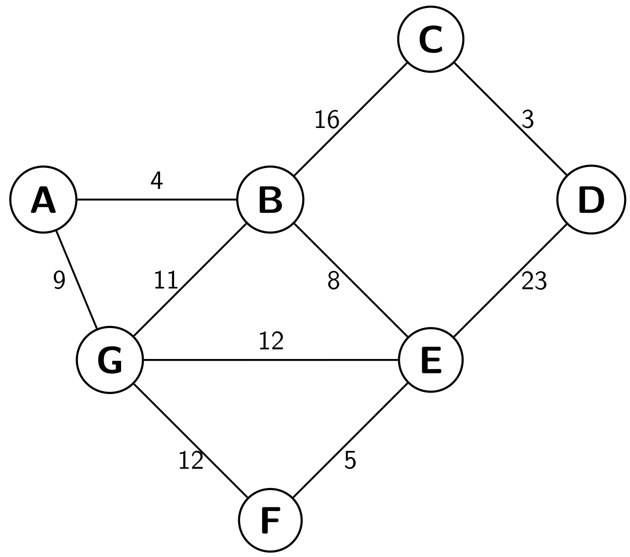
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



11. Now there are $[V(G) = 7)]$ *vertices* **and** $\mathcal{F}$ **has** $[V(G) - 1 = 6)]$ *edges*, **Kruskal's algorithm terminates and returns** $\mathcal{F}$**, which is our minimum-weight spanning tree shown in step (10) above.** $\square$

5 Standard 8- 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.

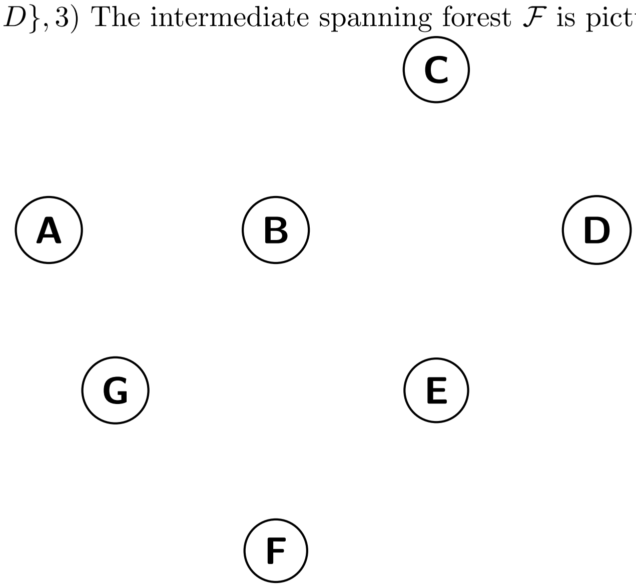


Answer:

1. 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 A.

$$Q = [(\{A, B\},\ 4),\ (\{A, G\},\ 9)]$$

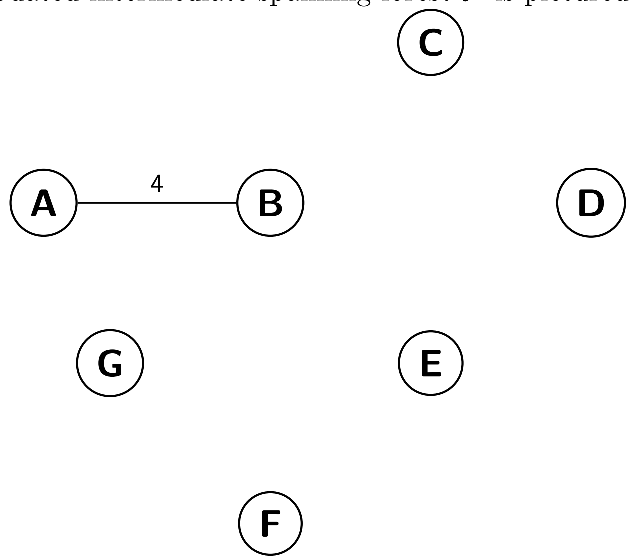
Here, $(\{C, D\}, 3)$ The intermediate spanning forest $\mathcal{F}$ is pictured below:



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

$$Q = [(\{B, E\},\ 8),\ (\{A, G\},\ 9),\ (\{B, G\},\ 11),\ (\{B, C\},\ 16)]$$

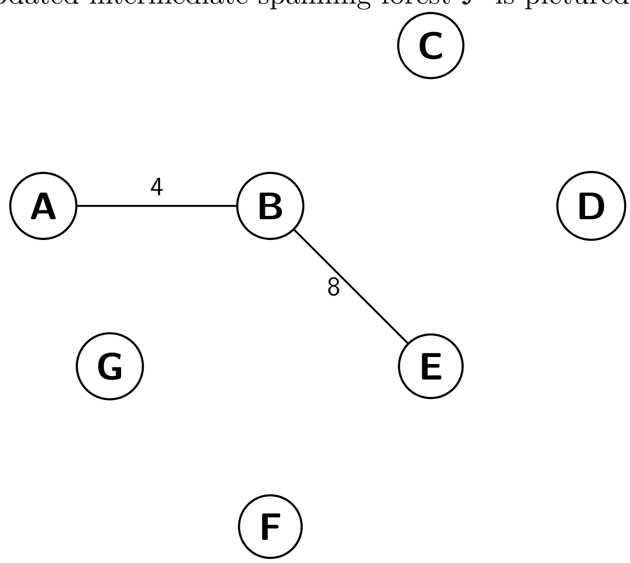
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



3. We poll the edge $\{B, E\}$ from the queue and mark $\{B, E\}$ as processed. Note that $w(\{B, E\}) = 8$. As $\{B, E\}$ has exactly one endpoint on the component containing A (which is the isolated vertex $\{A, B\}$), we add $\{B, E\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to E.

$$Q = [(\{E, F\}, 5), (\{A, G\}, 9), (\{B, G\}, 11), (\{E, G\}, 12), (\{B, C\}, 16), (\{E, D\}, 23)]$$

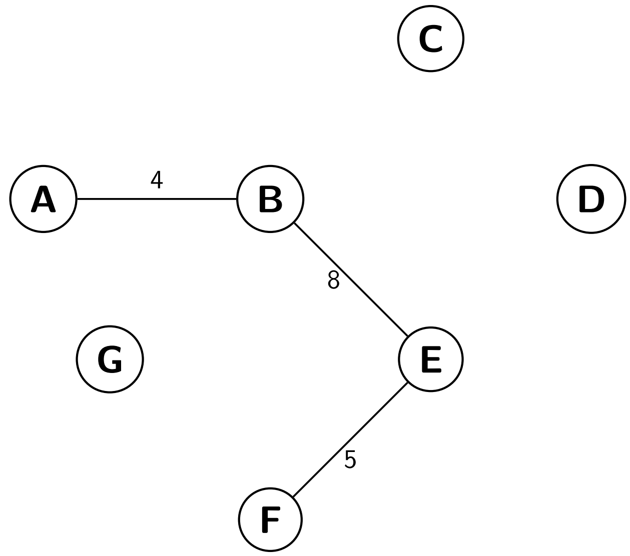
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



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

$$Q = [(\{A, G\}, 9), (\{B, G\}, 11), (\{F, G\}, 12), (\{E, G\}, 12), (\{B, C\}, 16), (\{E, D\}, 23)]$$

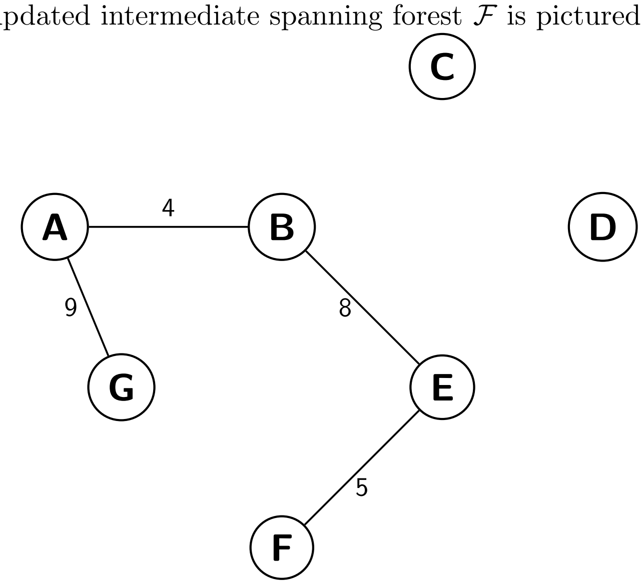
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



5. We poll the edge $\{A, G\}$ from the queue and mark $\{A, G\}$ as processed. Note that $w(\{A, G\}) = 9$. As $\{A, G\}$ has exactly one endpoint on the component containing A (which is the isolated vertex $\{A, B\}$), we add $\{A, G\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to C.

$$Q = [(\{C, D\}, 3), (\{B, G\}, 11), (\{F, G\}, 12), (\{E, G\}, 12), (\{B, C\}, 16), (\{E, D\}, 23)]$$

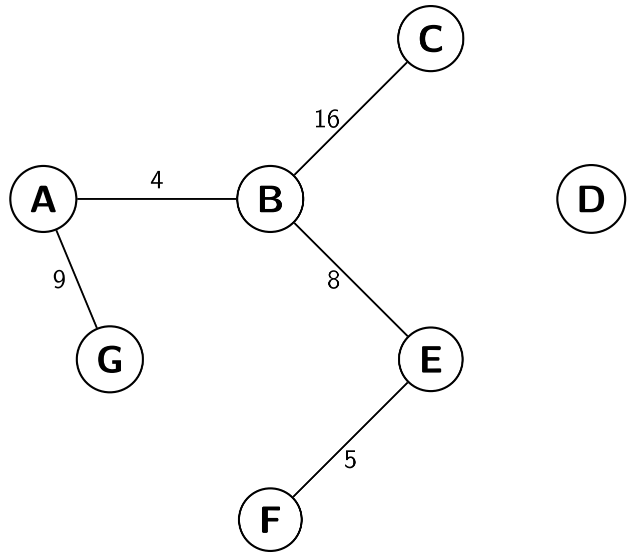
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



6. We poll the edge $\{B, C\}$ from the queue and mark $\{B, C\}$ as processed. Note that $w(\{B, C\}) = 16$. As $\{B, C\}$ has exactly one endpoint on the component containing A (which is the isolated vertex $\{E, B\}$), we add $\{C, B\}$ to $\mathcal{F}$. We then push into the priority queue the unprocessed edges incident to D.

$$Q = [(\{C, D\}, 3), (\{B, G\}, 11), (\{F, G\}, 12), (\{E, G\}, 12), (\{E, D\}, 23)]$$

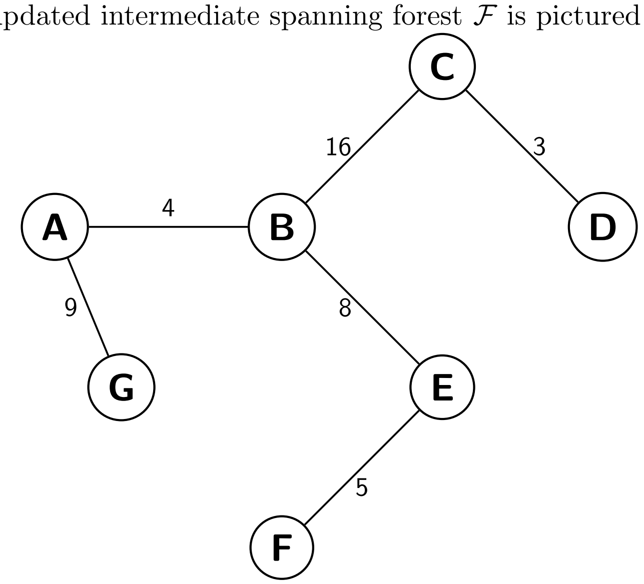
The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



7. We poll the edge $\{C, D\}$ from the queue and mark $\{C, D\}$ as processed. Note that $w(\{C, D\}) = 3$. As $\{C, D\}$ has exactly one endpoint on the component containing A (which is the isolated vertex $\{B, C\}$), We then push into the priority queue the unprocessed edges incident to D (provided said edges are not already in the priority queue).

$$Q = [(\{B, G\}, 11), (\{F, G\}, 12), (\{E, G\}, 12), (\{E, D\}, 23)]$$

The updated intermediate spanning forest $\mathcal{F}$ is pictured below:



8. Now $\mathcal{F}$ has $[V(G) - 1 = 7 - 1 = 6)]$ *edges*, **the algorithm terminates and returns $\mathcal{F}$, which is our minimum-weight spanning tree shown in step (7) above.** $\square$