CSCI 3104 Spring 2022
Instructor: Profs. Chen and Layer

Problem Set 3

Due Date .... **February 8, 2022**
Name .... **Julia Troni**
Student ID **109280095**
Collaborators .... **Null**

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)

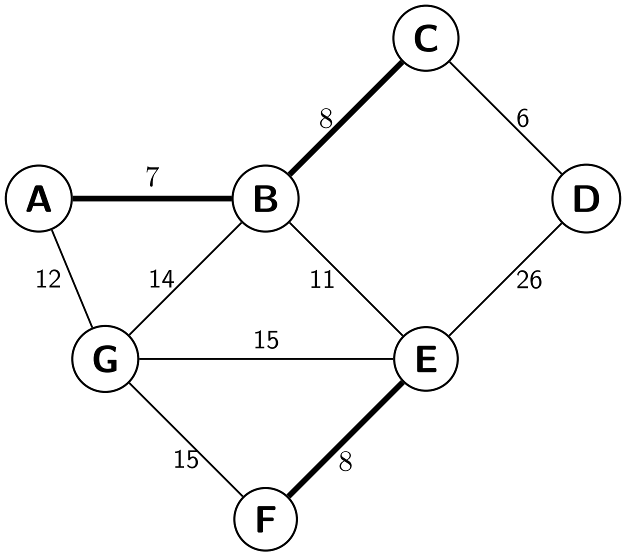
Problem 1.

- 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, Julia Troni. □

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.



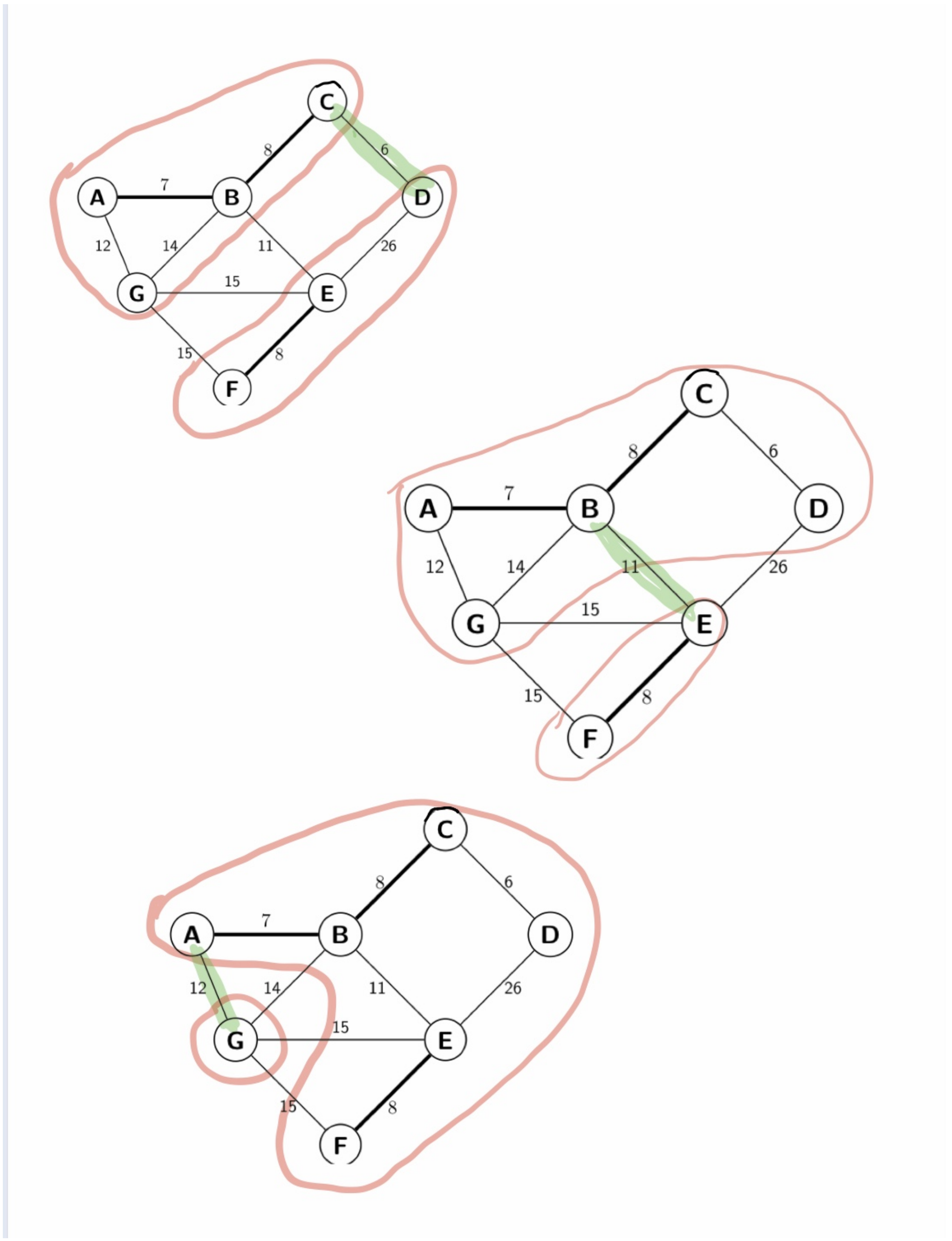
Answer. $\{C, D\}$, $\{B, E\}$, and $\{A, G\}$ are **safe** edges with respect to $\mathcal{F}$ because they are each the minimum weight edges that crosses a partition of the vertices (a cut), so they are each light edges. Since any light edge that connects two components in an intermediate spanning forest is safe, they are safe.
$\{B, G\}$, $\{G, E\}$, $\{G, F\}$, $\{D, E\}$ are **undecided** edges with respect to $\mathcal{F}$ because they are neither safe, nor useless There are no edges that will create a cycle with respect to $\mathcal{F}$ so there are no useless edges.

1. $\{C, D\}$ is the minimum weight edge with exactly one endpoint in the the component $\{A, B, C, G\}$ and the minimum weight edge with exactly one edge in the component $\{F, E, D\}$. Therefore $\{C, D\}$ is a light edge and hence a **safe** edge with respect to $\mathcal{F}$

2. $\{B, E\}$ is the minimum weight edge with exactly one endpoint in the the component $\{A, B, C, G\}$ and the minimum weight edge with exactly one edge in the component $\{F, E\}$. Therefore $\{B, E\}$ is a light edge and hence a **safe** edge with respect to $\mathcal{F}$

3. $\{A, G\}$ is the minimum weight edge with exactly one endpoint in the the component $\{A, B, C, D, E, F\}$ and the minimum weight edge with exactly one edge in the component $\{G\}$. Therefore $\{A, G\}$ is a light edge and hence a **safe** edge with respect to $\mathcal{F}$

   See below for a visual of the safe edges



4. While the edge $\{B, G\}$ has exactly one endpoint in the component $\{A, B, C, D, E, F\}$ and one endpoint in the component $\{G\}$, it is not a minimum weight edge in doing so. Therefore, $\{B, G\}$ is an **undecided** edge with respect to $\mathcal{F}$

5. Similarly the edge $\{G, E\}$ has exactly one endpoint in the component $\{A, B, C, D, E, F\}$ and one endpoint in the component G, but it is not a minimum weight edge in doing so. Therefore, $\{G, E\}$ is an **undecided** edge with respect to $\mathcal{F}$

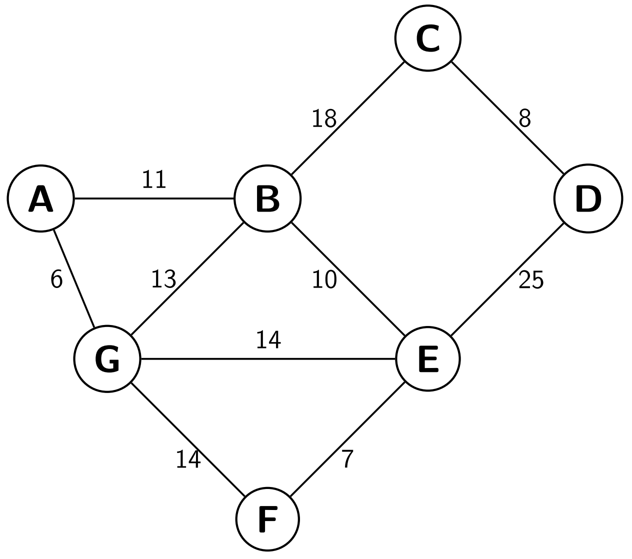
6. Similarly the edge $\{G, F\}$ has exactly one endpoint in the component $\{A, B, C, D, E, F\}$ and one endpoint in the component $\{G\}$, although it is not a minimum weight edge connecting these components. Therefore, $\{G, F\}$ is an **undecided** edge with respect to $\mathcal{F}$

7. Lastly, the edge $\{D, E\}$ has exactly one endpoint in the component $\{A, B, C, D, G\}$ and one endpoint in the component $\{E, F\}$, however it is not a minimum weight edge connecting these components. Therefore, $\{D, E\}$ is an **undecided** edge with respect to $\mathcal{F}$

$\square$

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. Kruskal's algorithm begins by initializing the intermediate spanning forest $\mathcal{F}$ to contain all the vertices but no edges. Then, it creates a priority queue in which it adds all of the edges of the graph, ordered from lowest weight to highest weight.
For this graph, the edges will be added to the queue such that initially
$Q = [(\{A, G\}, 6), (\{E, F\}, 7), (\{C, D\}, 8), (\{B, E\}, 10),$
$(\{A, B\}, 11), (\{B, G\}, 13), (\{E, G\}, 14), (\{F, G\}, 14), (\{B, C\}, 18), (\{D, E\}, 25)]$

- Poll from Q, which returns $(\{A, G\}, 6)$. Since A and G are on different components of $\mathcal{F}$, $\{A, G\}$ does not create a cycle so we add it to $\mathcal{F}$. Now the edges of $\mathcal{F}$ are $[(\{A, G\}, 6)]$ and
$Q = [(\{E, F\}, 7), (\{C, D\}, 8), (\{B, E\}, 10), (\{A, B\}, 11),$
$(\{B, G\}, 13), (\{E, G\}, 14), (\{F, G\}, 14), (\{B, C\}, 18), (\{D, E\}, 25)]$

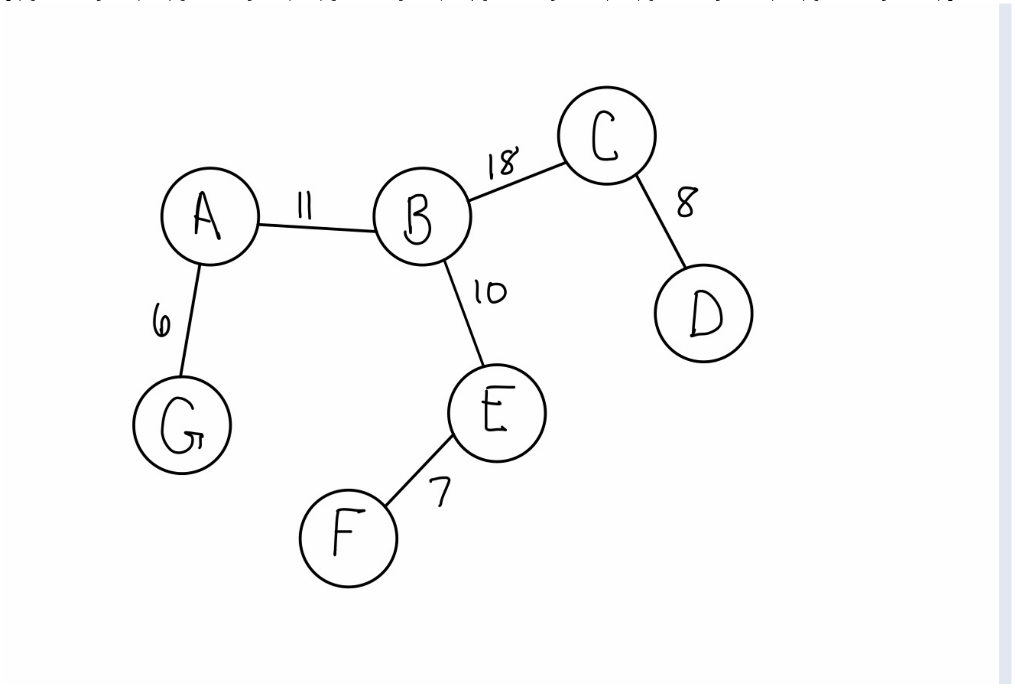
- Poll from Q, which returns $(\{E, F\}, 7)$. Since E and F are on different components of $\mathcal{F}$, adding that edge would not create a cycle so we add the edge to $\mathcal{F}$. Now the edges of $\mathcal{F}$ are $[(\{A, G\}, 6), (\{E, F\}, 7)]$ and
$Q = [(\{C, D\}, 8), (\{B, E\}, 10), (\{A, B\}, 11), (\{B, G\}, 13), (\{E, G\}, 14), (\{F, G\}, 14), (\{B, C\}, 18), (\{D, E\}, 25)]$

- Poll from Q, which returns $(\{C, D\}, 8)$. Since C and D are on different components of $\mathcal{F}$, adding that edge would not create a cycle so, we add the edge to $\mathcal{F}$. Now the edges of $\mathcal{F}$ are $[(\{A, G\}, 6), (\{E, F\}, 7), (\{C, D\}, 8)]$ and
$Q = [(\{B, E\}, 10), (\{A, B\}, 11), (\{B, G\}, 13), (\{E, G\}, 14), (\{F, G\}, 14), (\{B, C\}, 18), (\{D, E\}, 25)]$

The algorithm will continue in this method, polling each edge from the priority queue and checking if it creates a cycle. If it does NOT create a cycle, the edge is added to $\mathcal{F}$. However, if it does create a cycle, meaning the two vertices are on the same component of $\mathcal{F}$, then the edge will NOT be added to $\mathcal{F}$.
So $(\{B, E\}, 10)$ will be the next added edge, followed by $(\{A, B\}, 11)$ and lastly $(\{B, C\}, 18)$

The algorithm terminates when $\mathcal{F}$ spans all vertices, and at which point $\mathcal{F}$ has $|V(G)| - 1 = 7 - 1 = 6$ edges and is the MST of the graph.
For this graph, the MST $\mathcal{F}$ contains the following edges and is pictured below

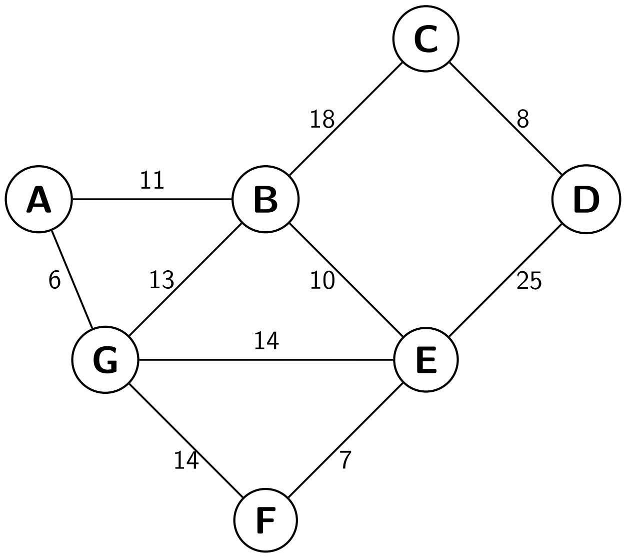
$[(\{A,G\},6),(\{E,F\},7),(\{C,D\},8),(\{B,E\},10),(\{A,B\},11),(\{B,C\},18)]$.



$\square$

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.

- Initialize the intermediate spanning forest, $\mathcal{F}$, to contain all of the vertices of G, but no edges. Then initialize the priority queue to contain the edges incident to the source vertex A. The priority queue will order the edges from lowest to highest weights.
  So at the start $Q = [(\{A, G\}, 6), (\{A, B\}, 11)]$

- Poll the edge $\{A, G\}$ from Q and mark $\{A, G\}$ as processed. $\{A, G\}$ has exactly one endpoint on the component containing A, hence it does not create a cycle, so we add $\{A, G\}$ to $\mathcal{F}$. Then push the unprocessed edges incident to G onto the queue.
  So $Q = [(\{A, B\}, 11), (\{G, B\}, 13), (\{G, E\}, 14), (\{G, F\}, 14)]$

- Poll the edge $\{A, B\}$ from Q and mark it as processed. $\{A, B\}$ has exactly one endpoint on the component containing A, so we add $\{A, B\}$ to $\mathcal{F}$. Then push the unprocessed edges incident to B (that are not already in Q) onto the queue.
  So $Q = [(\{B, E\}, 10), (\{G, B\}, 13), (\{G, E\}, 14), (\{G, F\}, 14), (\{B, C\}, 18)]$

- Now poll the edge $\{B, E\}$ from Q and mark it as processed. $\{B, E\}$ has exactly one endpoint on the component containing A, so we add $\{B, E\}$ to $\mathcal{F}$. Then push the unprocessed edges incident to E (that are not already in Q) onto the queue.
  So $Q = [(\{E, F\}, 7), (\{G, B\}, 13), (\{G, E\}, 14), (\{G, F\}, 14), (\{B, C\}, 18), (\{E, D\}, 25)]$

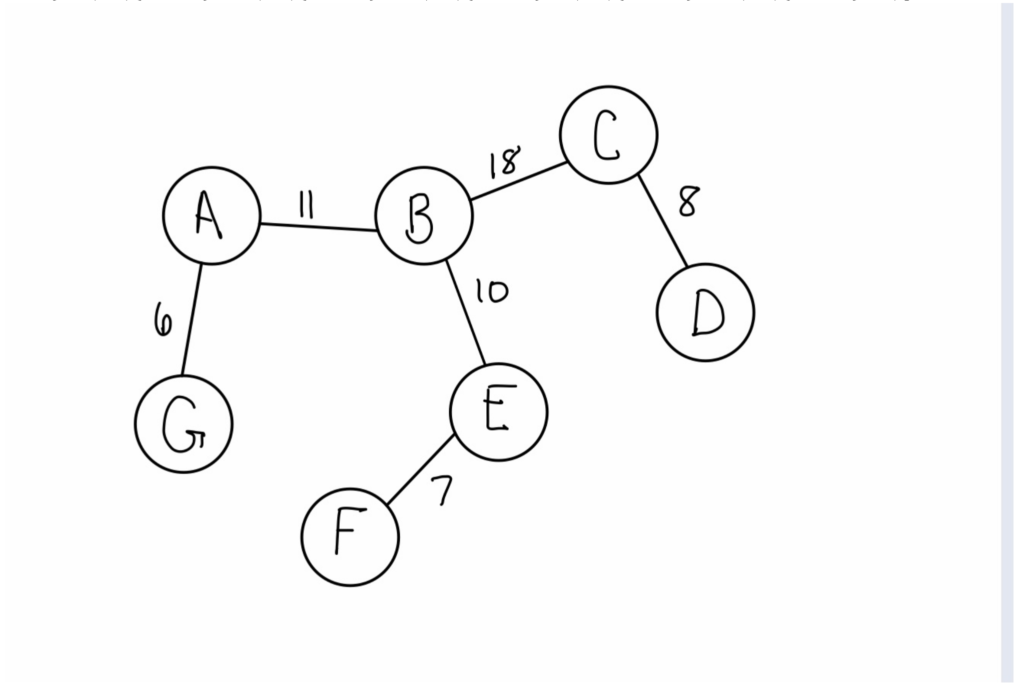
- Now poll the edge $\{E, F\}$ from Q and mark it as processed. $\{E, F\}$ has exactly one endpoint on the component containing A, so we add $\{E, F\}$ to $\mathcal{F}$. Since there are no edges incident to F that are not already in Q, we do not push any new edges.
  So $Q = [(\{G, B\}, 13), (\{G, E\}, 14), (\{G, F\}, 14), (\{B, C\}, 18), (\{E, D\}, 25)]$

- Now poll the edge $\{G, B\}$ from Q and mark it as processed. As both G and B belong to the component containing A, we do NOT add it to $\mathcal{F}$.
  Now $Q = [(\{G, E\}, 14), (\{G, F\}, 14), (\{B, C\}, 18), (\{E, D\}, 25)]$

- Now poll the edge $\{G, E\}$ from Q and mark it as processed. As both G and E belong to the component containing A, we do NOT add it to $\mathcal{F}$.
  Now $Q = [(\{G, F\}, 14), (\{B, C\}, 18), (\{E, D\}, 25)]$

- Now poll the edge $\{G, F\}$ from Q and mark it as processed. As both G and F belong to the component containing A, we do NOT add it to $\mathcal{F}$.
  Now $Q = [(\{B, C\}, 18), (\{E, D\}, 25)]$
- Now poll the edge $\{B, C\}$ from Q and mark it as processed. $\{B, C\}$ has exactly one endpoint on the component containing A, so we add $\{B, C\}$ to $\mathcal{F}$. Then push the unprocessed edges incident to C (that are not already in Q) onto the queue.
  Now $Q = [(\{C, D\}, 8), (\{E, D\}, 25)]$
- Now poll the edge $\{C, D\}$ from Q and mark it as processed. $\{C, D\}$ has exactly one endpoint on the component containing A, so we add $\{C, D\}$ to $\mathcal{F}$. Since there are no edges incident to D that are not already in Q, we do not push any new edges.
  Now $Q = [(\{E, D\}, 25)]$
- Now poll the edge $\{E, D\}$ from Q and mark it as processed. As both E and D belong to the component containing A, we do NOT add it to $\mathcal{F}$. Note: $\{E, D\}$ is the max weight edge, which will never be added to the MST
  Now $Q = []$ and $\mathcal{F}$ has $|V(G)| - 1 = 7 - 1 = 6$ edges so the algorithm terminates and returns $\mathcal{F}$ as shown below.

For this graph, the MST $\mathcal{F}$ contains the following edges added in this order
$[(\{A, G\}, 6), (\{A, B\}, 11), (\{B, E\}, 10), (\{E, F\}, 7), (\{B, C\}, 18), (\{C, D\}, 8)]$.



□