Homework 7
Due Friday November 9 at 11:59pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the \LaTeX template we provided, then you can make sure it appears by filling in the \texttt{yourname} command.

- This assignment is due Friday November 9 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.

- Solutions must be typeset in \LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.

- I encourage you to work with your classmates on the homework problems. However, you must write all solutions by yourself, in your own words. Do not share any written or typed solutions. Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the \texttt{yourcollaborators} command.

- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly forbidden.
Problem 1. MST Practice

Compute an MST in the following graph. You do not need to justify your answer. **Hint:** Draw your MST by filling in the edges on the skeleton provided.

Solution:
Problem 2. Shortest Paths Practice

Solve the single-source shortest path problem on the following graph, using node 1 as the source. Write the distance from $s$ to each node and write the parent of each node in the shortest path tree. **Hint:** Write your solution using the skeleton table provided.

![Graph Diagram]

Solution:

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>???</td>
<td>???</td>
<td>???</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>Parent</td>
<td>???</td>
<td>???</td>
<td>???</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>
**Problem 3. Cycle killer: qu’est-ce que c’est**

Suppose we are given an undirected, connected, weighted graph \( G = (V, E, \{w(e)\}) \), with non-negative edge weights \( w(e) \geq 0 \). In this problem we will design an algorithm to remove a set of edges \( F \subseteq E \) from the graph such that the resulting graph \( G' = (V, E \setminus F) \) is acyclic and the total weight of edges in \( F \) is as small as possible. Such a set is called a feedback edge set and this problem is sometime called the minimum weight feedback edge set problem.

(a) Design an algorithm for finding a maximum spanning tree of \( G \). **Hint:** Transform the graph from \( G \) to \( G' \) so that the minimum spanning tree of \( G' \) is a maximum spanning tree of \( G \).

   (i) Describe your algorithm in pseudocode.

   **Solution:**

   (ii) Justify that your algorithm is correct.

   **Solution:**

   (iii) Analyze your algorithm’s running time.

   **Solution:**

(b) Design an algorithm to compute a minimum weight feedback edge set of \( G \). **Hint:** Show that if \( F^* \) is a minimum weight feedback edge set, then \( E \setminus F^* \) is a spanning tree.

   (i) Describe your algorithm in pseudocode.

   **Solution:**

   (ii) Justify that your algorithm is correct.

   **Solution:**

   (iii) Analyze your algorithm’s running time.

   **Solution:**
**Problem 4. Anti-Kruskal**

In this problem we will see a new algorithm for finding an MST—the *anti-Kruskal* algorithm. Recall that Kruskal’s algorithm starts with $T = \emptyset$, considers edges $e$ in *ascending* order of weight, and adds $e$ to $T$ as long as doing so would not create a cycle. The *anti-Kruskal algorithm* starts with $T = E$, considers edges $e$ in *descending* order of weight, and removes $e$ from $T$ as long as doing so would not make $T$ disconnected.

Explain why the anti-Kruskal algorithm outputs an MST. You may assume that all edge-weights are distinct and you may use the cut and cycle properties of MSTs without proof.

**Solution:**