Graphs (7 points)

1. [3 points] Given an adjacency-list representation of a multigraph $G = (V, E)$, describe an $O(V + E)$-time algorithm to compute the adjacency-list representation of the "equivalent" undirected graph $G' = (V, E')$, where E' consists of the edges in E with all multiple edges between two vertices replaced by a single edge and with all self-loops removed.

> **Solution:** Create an array A of size $|V|$. For a list in the adjacency list corresponding to vertex v, examine items on the list one by one. If any item is equal to v, remove it. If vertex u appears on the list, examine $A[u]$. If it's not equal to v, set it equal to v. If it's equal to v, remove u from the list. since we have constant time lookup in the array, the total runtime is $O(V + E)$.

2. [4 points] The incidence matrix of a directed graph $G = (V, E)$ with no self-loops is a $|V| \times |E|$ matrix $B = (b_{ij})$ such that

$$b_{ij} = \begin{cases} -1 & \text{if edge } j \text{ leaves vertex } i \\ 1 & \text{if edge } j \text{ enters vertex } i \\ 0 & \text{otherwise} \end{cases}$$

Describe what the entries of the matrix product BB^T represent, where B^T is the transpose of B.

> **Solution:**
>
> $$BB^T(i,j) = \sum_{e \in E} b_{ie} b^T_{ej} = \sum_{e \in E} b_{ie} b_{je}$$
>
> - If $i = j$, then $b_{ie} b_{je} = 1$ (it is $1 \cdot 1$ or (-1) $\cdot$ (-1)) whenever e enters or leaves vertex i, and 0 otherwise.
> - If $i \neq j$, then $b_{ie} b_{je} = -1$ when $e = (i, j)$ or $e = (j, i)$, and 0 otherwise.
>
> Thus,
>
> $$BB^T(i,j) = \begin{cases} \text{degree of } i = \text{in-degree } + \text{ out-degree} & \text{if } i = j \\ -(\# \text{ of edges connecting } i \text{ and } j) & \text{if } i \neq j \end{cases}$$

Minimum Spanning Trees (13 points)

3. [2 points] Professor Sabatier conjectures the following converse of Theorem 23.1. Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let $(S, V - S)$ be any cut of G that respects A, and let (u, v) be a safe edge for A crossing $(S, V - S)$. Then, (u, v) is a light edge for the cut. Show that the professor's conjecture is incorrect by giving a counterexample.

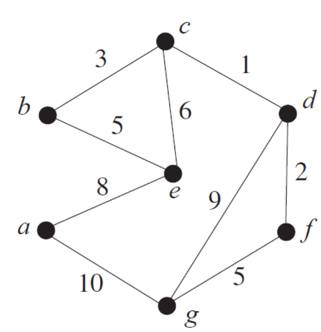
Solution: Let G be the graph with 4 vertices: u, v, w, z. Let the edges of the graph be $(u, v), (u, w), (w, z)$ with weights $3, 1$, and 2 respectively. Suppose A is the set $\{(u, w)\}$. Let $S = A$. Then S clearly respects A. Since G is a tree, its minimum spanning tree is itself, so A is trivially a subset of a minimum spanning tree. Moreover, every edge is safe. In particular, (u, v) is safe but not a light edge for the cut. Therefore Professor Sabatier's conjecture is false.

4. [4 points] Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge crossing the cut. Show that the converse is not true by giving a counterexample.

Solution: Suppose that for every cut of the graph there is a unique light edge crossing the cut, but that the graph has 2 spanning trees T and T'. Since T and T' and T are distinct, there must exist edges (u, v) and (x, y) such that (u, v) is in T but not T' and (x, y) is in T' but not T. Let $S = \{u, x\}$. There is a unique light edge which spans this cut. Without loss of generality, suppose that it is not (u, v). Then we can replace (u, v) by this edge in T to obtain a spanning tree of strictly smaller weight, a contradiction. Thus the spanning tree is unique.

For a counter example to the converse, let $G = (V, E)$ where $V = \{x, y, z\}$ and $E = \{(x, y), (y, z), (x, z)\}$ with weights $1, 2$, and 1 respectively. The unique minimum spanning tree consists of the two edges of weight 1, however the cut where $S = \{x\}$ doesn't have a unique light edge which crosses it, since both of them have weight 1.

5. [3 points] Consider the following weighted graph:



Use Kruskal's algorithm to find a minimum spanning tree for the graph, and indicate the order in which edges are added to form the tree.

Solution: Order of adding the edges: $\{c, d\}, \{d, f\}, \{b, c\}, \{b, e\}, \{f, g\}, \{a, e\}$

or $\{c, d\}, \{d, f\}, \{b, c\}, \{f, g\}, \{b, e\}, \{a, e\}$

6. [4 points] Let $G = (V, E)$ be any weighted connected graph. If C is any cycle of G, then show (formally) that the heaviest edge of C (i.e., the edge with the largest weight) cannot belong to a minimum spanning tree of G. (Assume that the heaviest edge is unique.) (Hint: use proof by contradiction; also, see the proof of Theorem 23.1).

Solution: Suppose the minimum spanning tree T includes the heaviest edge of the cycle $C = \langle c_0, c_1, \cdots, c_k \rangle$. Suppose the heaviest edge is (c_0, c_1). Form T' by removing (c_0, c_1) from T. T' is disconnected and c_0 and c_1 are in different components. Start from c_1 and continue by considering $c_2, c_3, \cdots, c_k$ and then c_0, find the first vertex (denote it by c_i) that is not in the same component as c_1 (we will have such vertex because there is at least one vertex that is not in the same component as c_1: c_0). Now add (c_{i-1}, c_i) (if $i = 0$, add (c_k, c_0)) to T'. T' is connected now and hence is a tree. However, $w(c_{i-1}, c_i) < w(c_0, c_1)$. So, $w(T') = w(T) - w(c_0, c_1) + w(c_{i-1}, c_i) < w(T)$ which is a contradiction with T being a minimum spanning tree. Therefore, the first assumption was incorrect and T does not include the heaviest edge of C.