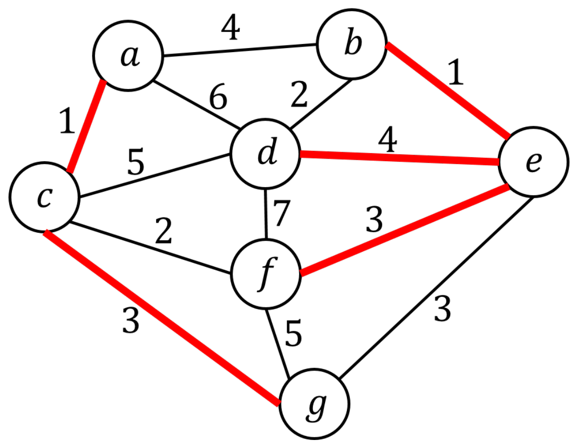
COSC 311: ALGORITHMS
MINI 6
Solutions

1. Definitions. Here's a graph: The thick red edges represent a subset of the graph's edges.



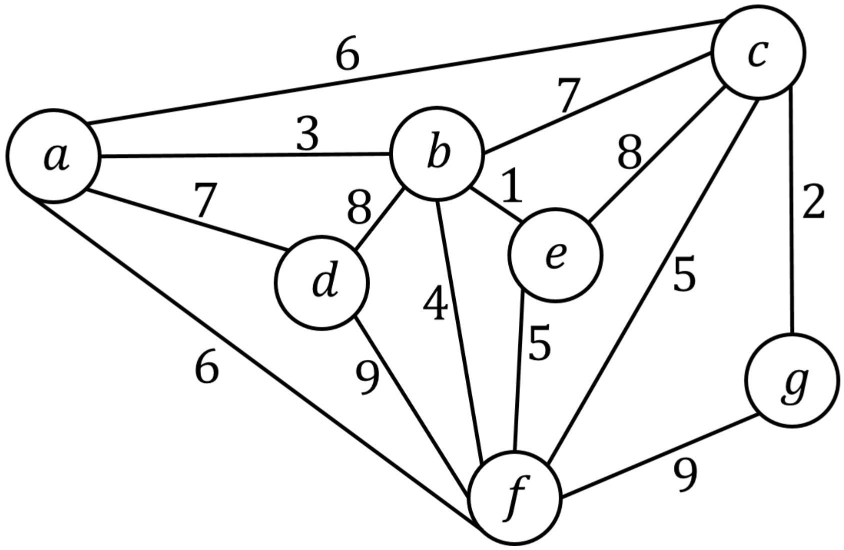
(a) Give a cut that respects the set of red edges.

Solution: Put $\{a, c, g\}$ on one side of the cut and $\{b, d, e, f\}$ on the other.

(b) List all the light edges across the cut you gave in part (a).

Solution: The edges crossing this cut are (a, b), (a, d), (c, d), (c, f), (e, g), and (f, g). Of these, (c, f) is a light edge because its weight is 2—lower than that of any other edge crossing the cut.

2. Minimum spanning trees. Here's a graph:



In what order are edges added to a minimum spanning tree when running each of the following algorithms? (You can break ties however you want, if needed.)

(a) Prim's algorithm

Solution: Prim's algorithm selects any vertex to begin with, and then at each step of the algorithm selects the lowest weight edge between a vertex in the tree and a vertex not in the tree. Let V be our set of vertices in the tree so far. Then at each step we will consider all edges in the graph which are connected to any vertex in V. Start with $V = \{a\}$. Then running Prim's algorithm does the following:
1. Min weight is $3 = (a, b)$. $V = \{a, b\}$.
2. Min weight is $1 = (b, e)$. $V = \{a, b, e\}$.
3. Min weight is $4 = (b, f)$. $V = \{a, b, e, f\}$.
4. Min weight is $5 = (e, f)$ and (c, f); but adding the edge (e, f) would create a cycle in my graph, violating the tree property. Therefore choose (c, f). $V = \{a, b, c, e, f\}$.
5. Min weight is $2 = (c, g)$. $V = \{a, b, c, e, f, g\}$.
6. The only vertex left to add is d, and the minimum weight edge which connects d to any other vertex is $7 = (a, d)$. Now $V = \{a, b, c, d, e, f, g\}$. Note: there are several other edges that have weight less than or equal to 7, but adding any of them will create a cycle in our graph which we cannot have.

Overall the edges were added in the following order: (a, b); (b, e); (b, f); (c, f); (c, g); (a, d)

(b) Kruskal's algorithm

Solution: In Kruskal's algorithm, we consider all edges in the entire graph at every step, and take the minimum edge that does not create a cycle. Again let V be our set of vertices added to the tree so far. Then running Kruskal's works as follows:
1. Min weight is $1 = (b, e)$. $V = \{b, e\}$.
2. Min weight is $2 = (c, g)$. $V = \{b, c, e, g\}$.
3. Min weight is $3 = (a, b)$. $V = \{a, b, c, e, g\}$.
4. Min weight is $4 = (b, f)$. $V = \{a, b, c, e, f\}$.
5. Min weight is $5 = (e, f)$ and (c, f), but again we cannot add the edge (e, f) since doing so would create a cycle. Therefore select (c, f). $V = \{a, b, c, e, f\}$.
6. Again we now have every vertex in our graph except for d, and we must choose the minimum weight edge to connect d to the graph, which is $7 = (a, d)$. Now $V = \{a, b, c, e, f\}$. Note: there are several other edges that have weight less than or equal to 7, but adding any of them will create a cycle in our graph which we cannot have.

Overall the set of edges were added in the following order: (b, e); (c, g); (a, b); (b, f); (c, f); (a, d). Note that this is the same set of edges that we chose while running Prim's, but is in a different order. However, it is not always the case that we will choose the same edges running either algorithm, and we also may sometimes choose the same edges in the same order. It all depends on the graph.