In the Zybooks reading, we saw Kruskal’s Algorithm. On the graph below and on the left, we are in the process of running Kruskal’s algorithm. Bolded edges have been selected to be kept.

**Question 1.** Is edge AD in any valid MST of the full graph? Why or why not? Can you give an argument for this that does not depend on the validity of Kruskal’s algorithm?

**Question 2.** How about edge DH? Does the same argument from the previous question apply to this?

**Question 3.** Will edge IL be added to any valid MST for this graph? Why or why not? Can you give an argument for this that does not depend on the validity of Kruskal’s algorithm?

**Question 4.** For the above-right graph, select a starting vertex and trace Jarnik’s algorithm. At each stage, identify the cut being examined and justify the edge selection.
This directed graph is a **prerequisite graph**.

An edge \((u, v)\) indicates you must take class \(u\) before you take class \(v\). What would the neighbor set \(N(u)\) indicate?

Suppose your friend wants to minor in ICS and choose the following classes listed in the graph above. They have ten quarters left, and have room to take one course per quarter. Give a valid ordering in which they can complete the minor at one class per quarter, while respecting prerequisite requirements.

\(\deg^{-}(v)\) is the in-degree of vertex \(v\), indicating the number of edges with \(v\) as their destination.

\(\deg^{+}(v)\) is the out-degree of vertex \(v\), indicating the number of edges with \(v\) as their start point.

\[\sum_{v \in V} \deg^{-}(v) = \sum_{v \in V} \deg^{+}(v) = |E|\]

**Topological Sort**

A *topological order* of a directed graph is an ordering of the vertices such that, if \(v_i\) appears before \(v_j\), then there is no path from \(v_j\) to \(v_i\) in the graph. For example, on the above graph, 51 cannot be before 31 or 6B.

**Question 5.** What must be true about the graph in order for it to have a topological order?

**Question 6.** How do we find a topological order *on paper*?

**Question 7.** How do we find a topological order by computer?