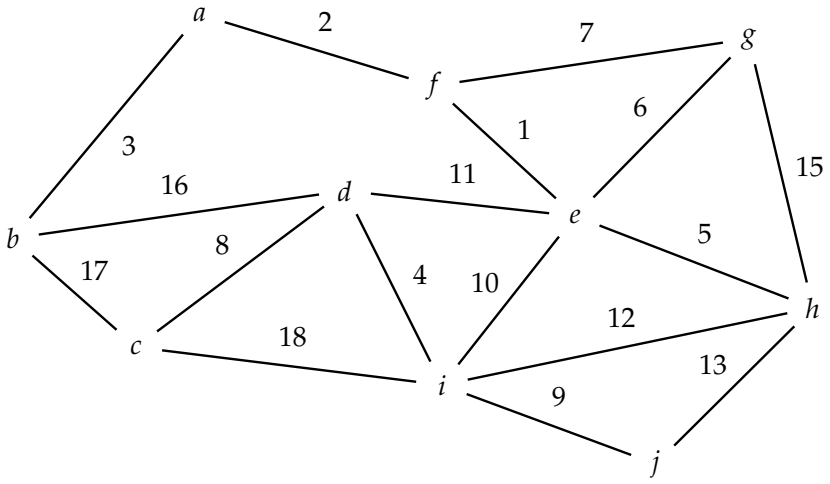
Algorithms: Minimum Spanning Trees

Model 1: Minimum-weight Spanning Subgraphs (20 mins)

Definition 1. Given an undirected graph $G = (V, E)$, a *spanning subgraph* of G is a connected subgraph $G' = (V, E')$ of G; that is, a connected graph with the same vertices as G and a subset of its edges.

Definition 2. Given a *weighted*, undirected graph G, the *minimum-weight spanning subgraph* (MWSS) of G is the spanning subgraph of G whose total weight (*i.e.* the sum of the weights of all its edges) is as small as possible.



1 Draw any spanning subgraph of the example graph from Model 1.

2 Can a graph G have more than one spanning subgraph?

3 Is it possible for a graph G to be a spanning subgraph of itself? Why or why not?

4 Is it possible for a disconnected graph to have a spanning subgraph? Why or why not?

5 Can a graph G have more than one minimum-weight spanning subgraph? Give an example, or explain why it is not possible.

6 Find a MWSS for the graph in Model 1. Does it have more than one?

Note: don't spend too much time on this question; just find a spanning subgraph which you reasonably think is the minimum, and move on.

7 Which of the following scenarios could be modeled by finding a MWSS?

(a) A railroad company wants to connect a given set of cities by train routes as cheaply as possible. Connecting two cities by a route costs an amount of money proportional to the distance between them.

(b) A company wants to network a set of data centers with high-speed fiber optic connections, and they want to mimimize the total amount of fiber optic cable used. There must be at least two routes between any pair of data centers, so that any one fiber optic link going down will not disconnect the network.

(c) Given a network of train routes between a collection of cities and the cost of each route, find the cheapest route between two given cities.

(d) You have the latest Megazorx puzzle toy, which can be in any one of a number of different states. There are various moves which can transform the Megazorx from one state to another. In order to impress your friends, you want to be able to transform it from any starting state into any other requested state (which may in general require a whole sequence of moves). You want to be able to do this

(i) ... using the fewest number of moves possible for each given pair of start and end states.

(ii) ... without having to memorize any more moves than absolutely necessary.

8 (Review) What is the definition of a *tree* graph?

9 Prove: in a weighted, undirected graph with positive weights, a MWSS must always be a tree.

Hint: use a proof by contradiction.

Because of the result from Question 9, we typically refer to a minimum-weight spanning subgraph as a *minimum spanning tree* (MST). Now that you understand the definition of a MST and some scenarios that MSTs can be used to model, let's explore some algorithms for finding them.

Model 2: Four algorithms (15 mins)

Given a weighted, undirected graph G, the goal of each algorithm is to pick a subset of the edges of G which constitute a MST.

- (**Kruskal's**) Consider all the edges in order from smallest to biggest weight; for each edge, pick it if and only if it does not complete a cycle with previously chosen edges.
- (**Prim's**) Choose an arbitrary vertex to start and mark it as visited. At each step, pick the smallest edge which connects any visited vertex to any unvisited vertex and which would not complete a cycle with previously chosen edges; mark the other end of the edge visited.
- (**Johnson's**) Choose any vertex to start. At each step, pick the smallest edge connected to the current vertex which has not already been chosen and would not complete a cycle with previously chosen edges. The other end of the edge becomes the new current vertex. Repeat until running out of options; then pick a new starting vertex and repeat the entire process, until all vertices are connected.
- (**Reverse delete**) Start with all edges initially "picked", and consider them in order from biggest to smallest weight. For each edge, throw it out (*i.e.* "unpick" it) if and only if doing so would not disconnect the remaining edges.

10 On the next page are four copies of the same graph. Use these to trace the execution of each algorithm in the model.

11 One of these algorithms is fake. Which one? Give an example showing why it does not work.