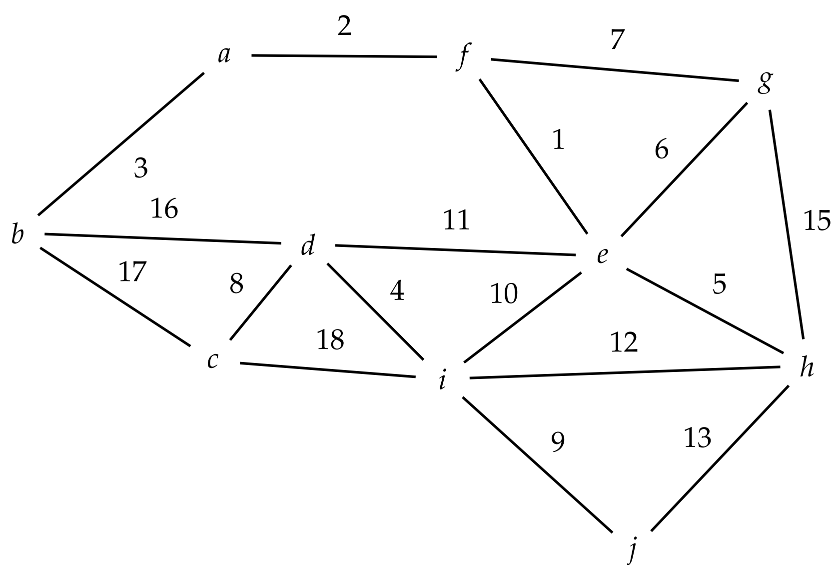
# *Algorithms: Kruskal's Algorithm*

---

In the previous activity you learned about minimum spanning trees and experimented with several different algorithms for finding them. In today's activity we will focus on Kruskal's Algorithm and prove that it works correctly.

## *Model 1: Kruskal's Algorithm (12 mins)*




---

**Require:** Undirected, weighted graph $G = (V, E)$

$T \leftarrow \varnothing$ ▷ $T$ holds the set of edges in the MST

Sort $E$ from smallest to biggest weight

**for** each edge $e \in E$ **do**

  **if** $e$ does not make a cycle with other edges in $T$ **then**

    Add $e$ to $T$

---

1 Simulate Kruskal's Algorithm on the graph in Model 1. What is the total weight of the resulting spanning tree?

2 The way the algorithm is written in Model 1, one must iterate through every single edge in $E$. However, this is not always neces-

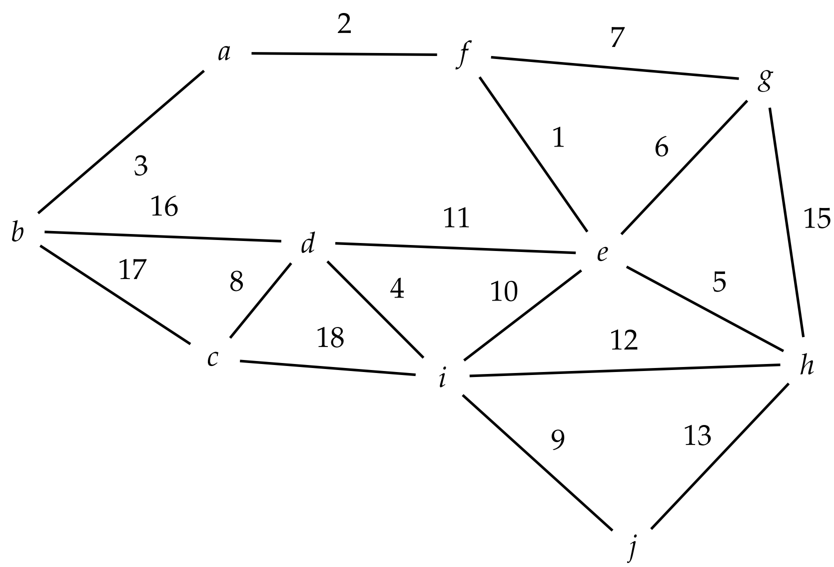
sary. Can you think of a simple way to tell when we can stop the loop early?

3 Explain why even in the worst case, $\Theta(\lg V) = \Theta(\lg E)$ in any graph. (Hint: what is the biggest $E$ can be, relative to $V$?)

4 In the above algorithm, how long does line 2 take? Simplify your answer using the observation from the previous question.

5 Can you think of a way to implement line 4? How long would it take?

*Model 2: The Cut Property (20 mins)*




**Definition 1.** A *cut* in a graph $G = (V, E)$ is a partition of the vertices $V$ into two sets $S$ and $T$, that is, every vertex is in either $S$ or $T$ but not both. We say that an edge $e$ *crosses* the cut $(S, T)$ if one vertex of $e$ is in $S$ and the other is in $T$.

**Theorem 2** (Cut Property). *Given a weighted, undirected graph $G = (V, E)$, let $S$ and $T$ be any partition of $V$, and suppose $e$ is some edge crossing the $(S, T)$ cut, such that the weight of $e$ is strictly smaller than the weight of any other edge crossing the $(S, T)$ cut. Then every minimum spanning tree of $G$ must include $e$.*

6 Give three examples of cuts in the graph from Model 2 and identify the smallest edge crossing each cut.

Let's prove the cut property.

*Proof.* Let $G$ be a weighted, undirected graph $G = (V, E)$, let $S$ and $T$ be an arbitrary partition of $V$ into two sets, and suppose $e = (x, y)$ is the smallest-weight edge with one endpoint in $S$ and one in $T$. We wish to show that ________________________________________.

We will prove the contrapositive. Suppose $M$ is a spanning tree of $G$ which does **not** contain the edge $e$. Since $M$ is a ______ it contains a unique ________________________________________

between any two ____________. So consider the unique ____________

in $M$ between ____________. It must cross the cut at least once since

*Hint*: draw a picture!

____________________________; suppose it crosses at $e' = (x', y')$, with $x' \in X$ and $y' \in Y$. We know that the weight of $e$ is smaller than

the weight of $e'$, since __________________________________.

Now take $M$ and replace ____________ with ____________; the result is

still ____________________ because ____________________,

but it has a smaller total ______________ because ______________.

So, we have shown that any spanning tree $M$ which does not

contain the edge $e$ can be made into a ________________________,

which means that $M$ is not a ______________. □

The cut property can be used to directly show the correctness of several MST algorithms. Let's prove the correctness of Kruskal's Algorithm; the proofs for the other algorithms are similar.

**Theorem 3.** *Kruskal's Algorithm is correct.*

*Proof.* Suppose at some step the algorithm picks the edge $e = (x, y)$. Let $X$ be the set of vertices connected to $x$ by edges which have been picked so far (not including $e$), and let $Y$ be all other vertices. $x \in X$ by definition. We know that $y \notin X$ since if it was, $e$ would

make a __________ but then Kruskal's Algorithm wouldn't __________.

$e$ therefore crosses the cut $(X, Y)$. No other edges which have been

picked previously cross the cut, since ________________________.

Therefore $e$ must be the smallest __________________________

because ________________________________________.

Therefore by the Cut Property $e$ must be in any MST and Kruskal's Algorithm is correct to pick it. □