1. Draw an example of a weighted graph which has $2$ MSTs.

There are lots of these. For example, take a triangle whose edges have values $1, 2, 2$. Then there are $2$ MSTs, obtained by dropping either of the “2”-edges.

2. Let $G, W$ be a weighted graph in which all edge weights are distinct.

Prove that the MST of $G, W$ is unique.

Proof: By contradiction.

Suppose there are two different MSTs, $T_1$ and $T_2$. Choose $(u, v) \in T_1 \setminus T_2$. Let $T_1(u), T_1(v)$ be the two sub-MSTs obtained by deleting $(u, v)$. Consider the path $p_{u,v}(T_2)$ between $u$ and $v$ in $T_2$. Let $E'$ be the edges of $p_{u,v}(T_2)$ that cross the cut $T_1(u), T_1(v)$. We are guaranteed that $|E'| \geq 1$.

Two cases:

(1) Suppose $W(e) > W(u, v)$ for some $e \in E'$. Then define $T_2' = T_2 \setminus \{e\} \cup \{(u, v)\}$. This is a spanning tree of cost strictly less than $T_2$. Contradiction!

(2) Suppose $\exists e \in E'$ with $W(e) < W(u, v)$. Then define $T_1' = (T_1 \setminus \{(u, v)\}) \cup \{e\}$. This is a spanning tree of $G$ with cost strictly less than $T_1$. Contradiction!

In either case we prove that one of $T_1, T_2$ was not a MST.

3. This question asks us to consider a modification to Prim’s algorithm where we add all the competing minimum-weight fringe edges in a single iteration of Prim’s algorithm. The question being asked is whether this strategy would lead to the construction of a Minimum Spanning Tree.

The answer is no.

Observation 1: This modification will not necessarily maintain a Tree through the life of the algorithm - for example, if the current tree $T$ contains nodes $a, b, c$, and there are three competing minimum fringe edges $(a, d), (a, f), (c, d)$ all of weight $2$, then adding all of them would induce a cycle. However, this answer is fairly easy … suppose we tweak this algorithm to only add one fringe edge per fringe-vertex?

Well, then the answer is still no.

To see this consider the following graph as an example:
At the stage when the current spanning tree $T$ contains $a, b, c$, there are three fringe edges $(a, d), (c, d), (b, e)$. Two of those edges, $(c, d), (b, e)$, have the minimum fringe edge weight 3. Also the fringe vertices $(d$ and $e$) are distinct, so would both be added to give a complete spanning tree of weight 9. However, if we just added one of $(c, d)$ and $(b, d)$, then the weight 2 edge $(d, e)$ would become available at the next step, and we would end up with a tree of value 8.

4. Consider an arbitrary edge $(u, v)$ in a graph $G$. Give a $O(|V| + |E|)$ time algorithm to determine, for a given weighted graph $(G, W)$, and a given edge $(u, v)$ of $G$, whether $(u, v)$ belongs to some MST of $G$.

Tutors: might want to motivate students on this question (especially the 'proof' part which they don’t always love) by telling them that it uses similar ideas to the proof of Prim’s Alg.

Our answer will exploit the following observation.

**Observation:** There does not exist any MST $T$ of $G$ which includes $(u, v)$ iff there is a path $p$ between $u$ and $v$ in $G$ such that for every edge $e \in p$, $W(e) < (u, v)$.

**Proof** (of Observation).

(we are again assuming $G$ is connected)

$\Leftarrow$: Suppose the condition holds and there was an MST $T$ containing $(u, v)$. Then consider the path $p$ of our assumption - clearly not all the edges of $p$ lie in $T$ (otherwise we would have a cycle). Let $e$ be any edge of $p$ whose endpoints belong to different subtrees of $T$, when we disconnect $T$ by deleting $(u, v)$. Define $T' = (T \setminus \{(u, v)\}) \cup \{e\}$. Then $W(T') < W(T)$, contradicting the assumption that $T$ was an MST.

$\Rightarrow$: Suppose there is no MST containing $(u, v)$. Suppose there was no path like $p$. Then we could take any MST $T$ (which cannot contain $(u, v)$), and consider the path $p_{u,v}(T)$ between $u$ and $v$ in $T$. By our assumption about paths, there must be some edge $e \in p_{u,v}(T)$ such that $W(e) \geq W(u, v)$. Define $T' = T \setminus \{e\} \cup \{(u, v)\}$, and then $W(T') = W(T)$, hence $T'$ is also an MST (containing $(u, v)$). Contradiction! Hence there must be a path $p$ with the desired properties.

**QED**
Algorithm: We use the observation. We need to explore $G$ to see whether we can find a path $p$ between $u$ and $v$ with edge weights all strictly less than $W(u,v)$. We can explore the graph via breadth-first search, using max rather than $\sum$ as our measure of a path. This takes time $O(|E| + |V|)$. See the Informatics 2B notes for details of breadth-first search.