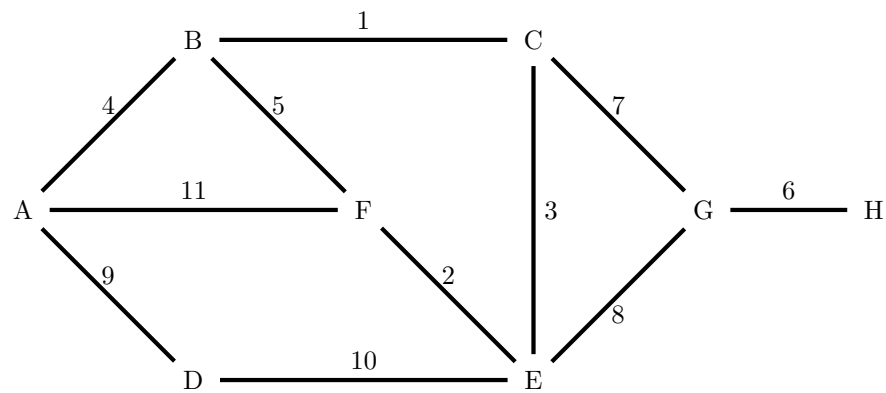
Layla Oesper, Jeff Ondich

One question of your choice (either #1 or #2 or #3) is due by **5:00PM Tuesday, April 30** and will be graded on completion only (1 point). Your full answers to all questions are due by **5:00PM Friday, May 3**. Each solution will be graded for correctness and clarity (4 points per question). Read the course information page/syllabus for further info about this 4-point scale.

0. Estimate the amount of time you spent on each question and include it at the top of your solution. Also, list your collaborators for each question at the top of your solution.

1. (a) For the graph shown below, run Kruskal's Algorithm by hand. List the order in which the edges get selected for inclusion in the resulting MST.

   (b) For the graph shown below, run Prim's Algorithm by hand starting with node F. List the order in which the edges get selected for inclusion in the resulting MST.



   (c) Show a graph for which there are (at least) two different runs of Prim's Algorithm that add the nodes and edges to the MST in different orders. List the sequence of edges added during each run.

2. For each of the following, determine whether the statement is true or false. If true, provide a short proof; if false, provide a counter-example. In each case, you start with a connected undirected graph $G = (V, E)$ and a weight function $w : E \longrightarrow \mathbb{Z}^{\geq 0}$.

   (a) Let $e \in E$ be the unique edge with the largest weight. Then e is not contained in any MST.

   (b) A *cut* is a pair $(S, V - S)$, where $S \subset V$, and both S and $V - S$ are nonempty. A *light edge* of a cut is an edge (u, v) where $u \in S$, $v \in V - S$, and $w(u, v)$ is the minimum among all such edges. (See the Cut Property 4.17 on page 145 of K&T for further discussion.) Let

$$L = \{(u, v) \in E \mid \exists \text{ a cut } (S, V - S) \text{ for which } (u, v) \text{ is a light edge}\}$$

   Then L is an MST.

3. Anybody who drives in city traffic during rush hour could be forgiven for thinking that there must be a better way. If only there were a type of vehicle that would take you where you want to go on a predictable schedule and conveniently located near your home–maybe you could even read or nap or watch videos about Prim's Algorithm during the trip!

   It turns out that the construction of commuter rail is political in the US. How political, you ask? From 2002 through 2023, it was illegal for Minnesota state and regional planning organizations to even study the possibility of passenger rail from Minneapolis to Northfield. It turns out that a lot of people don't like having trains going through their neighborhoods, and they elect legislators who agree with them.

Layla Oesper, Jeff Ondich

But now that the gag order has been lifted, the nonprofit public interest group *Better Living Through Graph Theory* has leapt into action! They have been studying what makes it possible to convince the legislature to approve a given *railway segment* (i.e., a stretch of rail between adjacent train stations). BLTGT has determined that the number of single-family homes within 100 yards of a rail segment (the segment's *house-count*) is negatively correlated with approval by the legislature. That is, the higher the house-count of a segment, the less likely the segment will get funded and built.

BLTGT has written to you, the most famous graph-theoretical consultant in the state. They've drawn up a railway map that will usher in an era of transportational utopia in Minnesota, but they're worried about getting it approved. So they want you to find a subset of the segments on their map that will connect all the stations to each other while minimizing the maximum house-count of those segments. Can you help?

More formally, suppose the BLTGT's map is a connected undirected graph $G = (V, E)$ with nodes representing stations and edges representing rail segments, and each segment has an associated house-count $h : E \longrightarrow \mathbb{Z}^{\geq 0}$. You seek a spanning tree $T = (V, E')$ with $E' \subseteq E$ such that $\max_{e \in E'} h(e)$ is as small as possible. Let's call such a tree a *minimum cranky homeowner tree* (MCHT).

(a) Prove that a minimum spanning tree of G *is* an MCHT of G.

Therefore we can conclude that finding an MCHT in $O(|E| \log |V|)$ time is straightforward, using any of the MST algorithms we've discussed. You can actually do better!

(b) Give a linear-time (that is, $O(|E| + |V|)$) algorithm that, given a graph G and an integer K, reports whether there is an MCHT whose maximum house-count is less than or equal to K. I'm not asking your algorithm to return the tree; just indicate whether or not it exists. Provide a proof of correctness, including that the runtime meets the specifications.

(c) Use your results from the previous question to give a linear-time algorithm to find an MCHT in a given graph G. You may find the following three items helpful.

- You may assume you have a subroutine that contracts a given set of edges in linear time. We'll discuss this in class.
- You may assume that it's possible (which it is) to find the median of a set S of integers in $O(|S|)$ time.
- Are you a fan of convergent infinite series? I am! Don't forget my old pal: $1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\ldots = 2$.