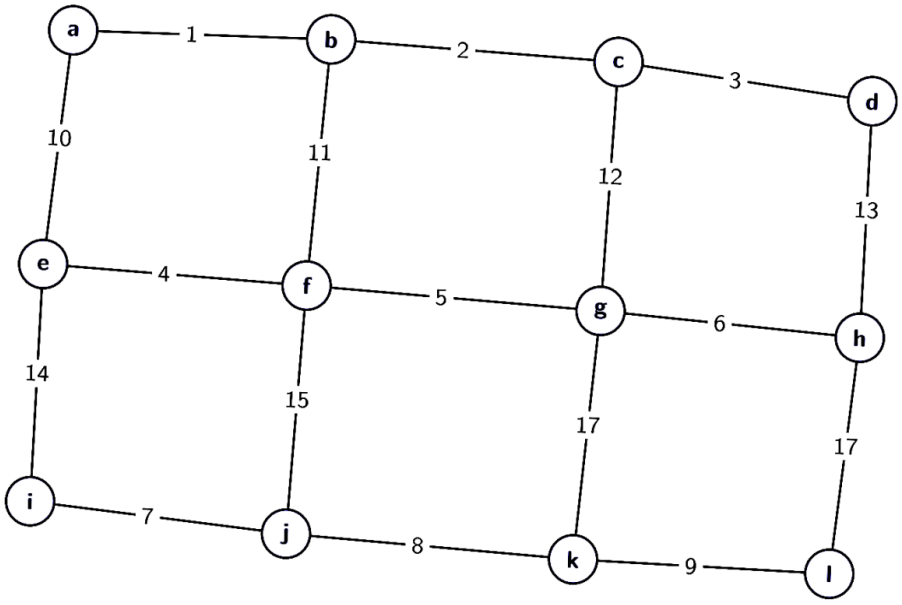
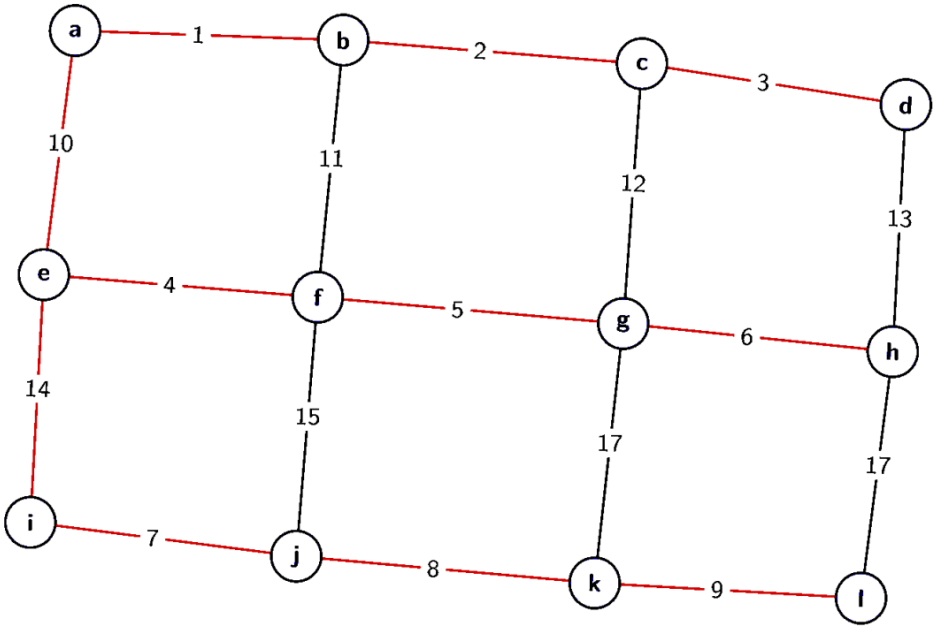
Section 10: P/NP, Final Review

3. LMNST!

Consider the following graph:



a) Find an MST of this graph using both of the two algorithms we've discussed in lecture. Make sure you say which algorithm you're using and show your work.



Using Prim's algorithm:

Vertex	Known	Cost of Edge
a	True	0
b	True	~~∞~~ 1
c	True	~~∞~~ 2
d	True	~~∞~~ 3
e	True	~~∞~~ 10
f	True	~~∞ 11~~ 4
g	True	~~∞ 12~~ 5
h	True	~~∞ 13~~ 6
i	True	~~∞~~ 14
j	True	~~∞ 15~~ 7
k	True	~~∞ 17~~ 8
l	True	~~∞ 17~~ 9

Using Kruskal's algorithm:
Starting Union Sets: {a}, {b}, {c}, {d}, {e}, {f}, {g}, {h}, {i}, {j}, {k}, {l}

Edge Being Processed	Resulting Union Find Forest
(a, b, 1)	{a, b}, {c}, {d}, {e}, {f}, {g}, {h}, {i}, {j}, {k}, {l}
(b, c, 2)	{a, b, c}, {d}, {e}, {f}, {g}, {h}, {i}, {j}, {k}, {l}
(c, d, 3)	{a, b, c, d}, {e}, {f}, {g}, {h}, {i}, {j}, {k}, {l}
(e, f, 4)	{a, b, c, d, e, f}, {g}, {h}, {i}, {j}, {k}, {l}
(f, g, 5)	{a, b, c, d, e, f, g}, {h}, {i}, {j}, {k}, {l}
(g, h, 6)	{a, b, c, d, e, f, g, h}, {i}, {j}, {k}, {l}

(i, j, 7)	{a, b, c, d, e, f, g, h}, {i, j}, {k}, {l}
(j, k, 8)	{a, b, c, d, e, f, g, h}, {i, j, k}, {l}
(k, l, 9)	{a, b, c, d, e, f, g, h}, {i, j, k, l}
(a, e, 10)	No change
(b, f, 11)	No change
(c, g, 12)	No change
(d, h, 13)	No change
(e, i, 14)	No change
(f, j, 15)	{a, b, c, d, e, f, g, h, i, j, k, l}

b) Using just the graph, how can you determine if it's possible that there are multiple MSTs of the graph? Does this graph have multiple MSTs?
A graph can only have multiple MSTs if it has multiple edges of the same weight. This graph has two 17s, but neither of them are used in the MST. So, there's only one MST here.

c) What is the asymptotic runtime of the algorithms that you used to compute the MSTs?
Prim's Algorithm takes $\mathcal{O}(|V| \lg(|V|) + |E| \lg(|V|))$, and Kruskal's Algorithm takes $\mathcal{O}(|E| \lg(|E|))$.

P, NP, NP-Complete

a) "NP" stands for:

Nondeterministic Polynomial

b) What does it mean for a problem to be in NP?

Given a candidate solution to a decision problem, we can verify whether the solution is correct in polynomial time.

c) For the following problems, circle ALL the sets they (most likely) belong to:

Is there a path of weight at most *k* from one vertex to another vertex in a weighted directed graph?

NP **P** NP-complete None of these

Is there a cycle that visits each edge in a graph exactly once?

NP **P** NP-complete None of these

Will this program run forever?

NP P NP-complete **None of these**

Can we find the prefix sum of an array in parallel using 10 processors?

NP **P** NP-complete None of these

Is there a path that starts and ends at the same vertex that visits every vertex exactly once?

NP P **NP-complete** None of these