Section 9: Graphs

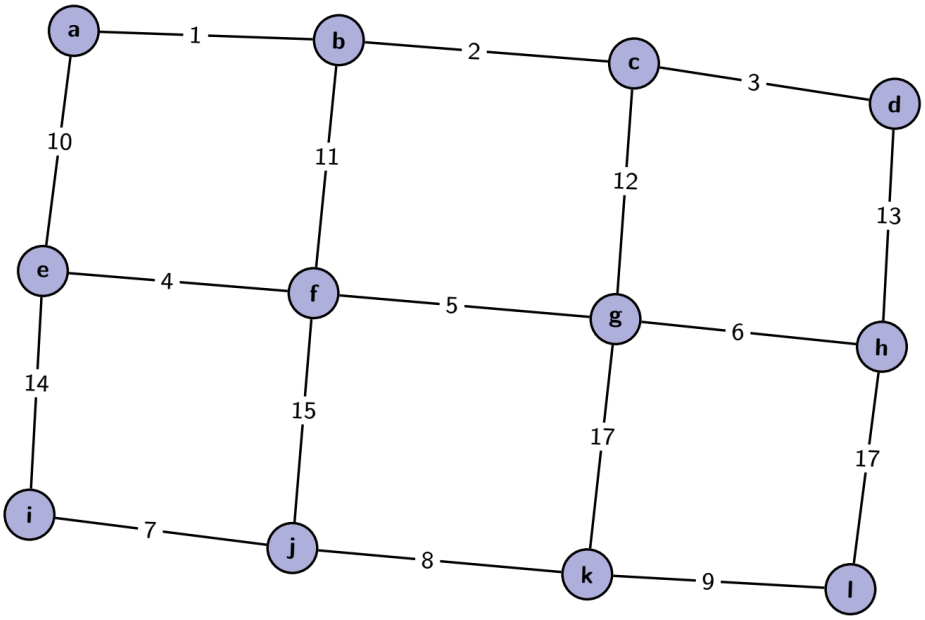
Recall Prim's Algorithm From Lecture:

```
PrimMST(Graph G)
   initialize distances to ∞
   mark source as distance 0
   mark all vertices unprocessed //and add to priority queue
   foreach(edge (source, v) ) {
      v.dist = weight(source,v)
   while(there are unprocessed vertices){
       let u be the closest unprocessed vertex //removeMin!
       add u.bestEdge to spanning tree
       foreach(edge (u,v) leaving u){
          if(weight(u,v) < v.cost){
             v.cost = weight(u,v) //updatePriority!!
             v.bestEdge = (u,v)
          }
       }
         mark u as processed
     }
   }
```

Running time: $\Theta(E \log V)$
Analysis same as Dijkstra, but can assume $E \geq V - 1$

1. 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.

Vertex	Known	Cost of Edge
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?

c) What is the asymptotic runtime of the algorithms that you used to compute the MSTs?