Scenario #1

You've made a new social networking app, Convrs. Users on Convrs can have "asymmetric" following (I can follow you, without you following me). You decide to allow people to form multi-user direct messages, but only if people are probably in similar social circles (to avoid spamming).

You'll allow a messaging channel to form only if for every pair of users a,b in the channel: a must follow b or follow someone who follows b or follow someone who follows someone who follows b, or …
And the same for b to a.

You'd like to be able to quickly check for any new proposed channel whether it meets this condition.

What are the vertices?

What are the edges?

What are we looking for?

What do we run?

Scenario #2

Sports fans often use the "transitive law" to predict sports outcomes -- In general, if you think A is better than B, and B is also better than C, then you expect that A is better than C.

Teams don't all play each other – from data of games that have been played, determine if the "transitive law" is realistic, or misleading about at least one outcome.

What are the vertices?

What are the edges?

What are we looking for?

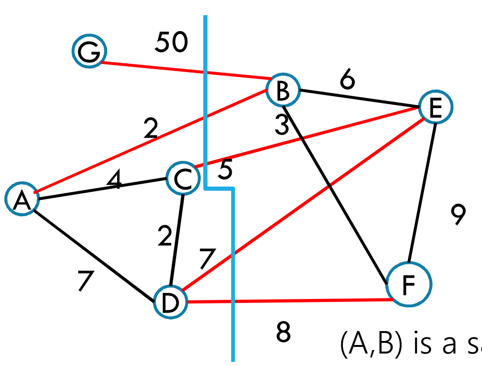
What do we run?

Prim's Algorithm

```
PrimMST(Graph G)
      initialize costToAdd to ∞
      mark source as costToAdd 0
      mark all vertices unprocessed, mark source as processed
      foreach(edge (source, v) ) {
             v.costToAdd = weight(source,v)
             v.bestEdge = (source,v)
      }
      while(there are unprocessed vertices){
             let u be the cheapest to add unprocessed vertex
             add u.bestEdge to spanning tree
             foreach(edge (u,v) leaving u){
                    if(weight(u,v) < v.costToAdd AND v not processed){
                           v.costToAdd = weight(u,v)
                           v.bestEdge = (u,v)
                    }
             }
     mark u as processed
      }
```

Safe Edge

Call an edge, e, a "**safe edge**" if there is some cut $(S, V \setminus S)$ where e is the minimum edge spanning that cut



(A,B) is a safe edge

(C,D) is a safe edge