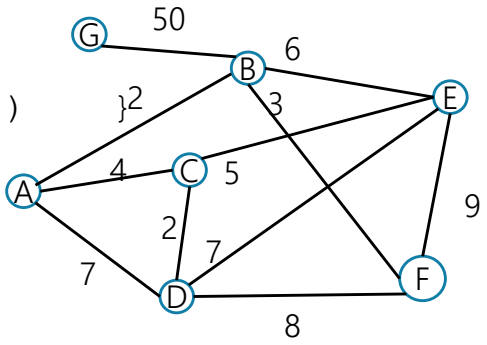
Try it Out

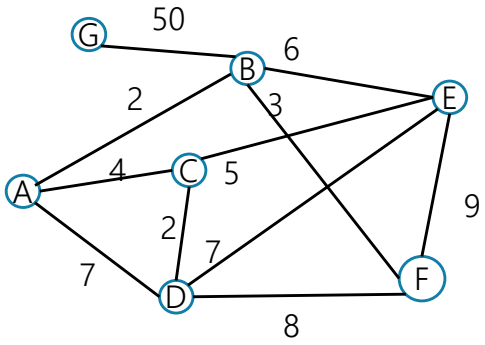
```
KruskalMST(Graph G)
  initialize each vertex to be a connected component
  sort the edges by weight
  foreach(edge (u, v) in sorted order){
    if(find(u) != find(v)){
      add (u,v) to the MST
      union(find(u),find(v))
  }
```



26

Try it Out

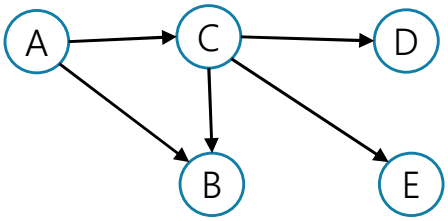
Edge	Include?	Reason



28

Ordering a DAG

Does this graph have a topological ordering? If so find one.



38

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?

53