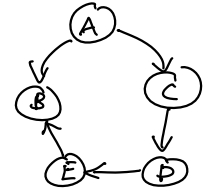


→ needed by



In what order should I take these courses so that I don't violate the prerequisites?

21, 35, 63, 31, 75
 21, 31, 35, 75, 63

● Not visited

● In-progress

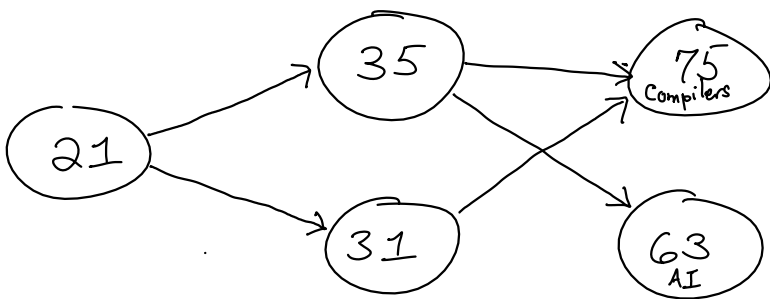
● Complete



Construct list of vertices such that each vertex appears after its predecessors (incoming edges' source vertices).

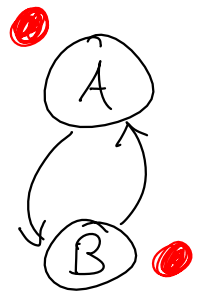
Useful for "dependency graphs"

Topological Sort



Function $\text{topoSort}(\text{Graph } g)$: \leftarrow returns list of vertices
 List<V> result \leftarrow new LL
 For each vertex v in g :
 $\text{visit}(g, v, \text{result})$
 Return result

Function $\text{visit}(\text{Graph } g, V \ v, \text{List}\langle V \rangle \ \text{result})$:
 If v has been marked complete: Return
 If v has been marked in-progress: \perp
 Mark v as in-progress
 For each neighbor n of v in g :
 $\text{visit}(g, n, \text{result})$
 Mark v as complete
 $\text{result.insertAtHead}(v)$



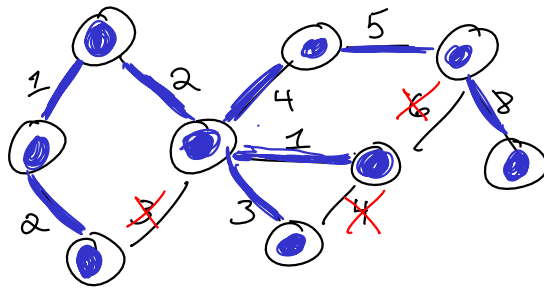
Minimum Spanning Trees

— what set of edges can I keep s.t. the total weight is minimal and graph is still connected?

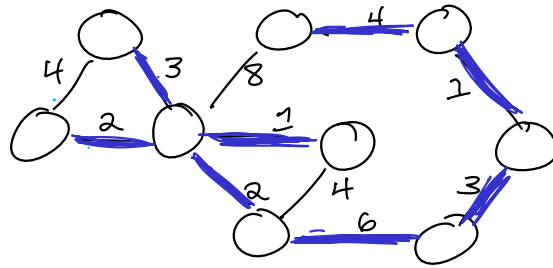
In a tree,

$$|E| = |V| - 1$$

Prim's
Algorithm



Kruskal's
Algorithm



Union-Find