



Data Structures with C++ : CS189

Lecture 11-2: Graphs Algorithms

Recap

- Topological Sort finds loops
 - Used to find process deadlocks
- DJ finds shortest/best cost path
 - For optimizing train or plane trips through multiple stations
 - A* is needed to become Google Maps
 - A* doesn't have to search the whole world, where DJ might
- Max Flow finds the max capacity of edges through a set of vertices
 - Used to improve network/water/electricity flow

Minimum Spanning Tree

- What is the smallest/cheapest number of edges I actually need to keep a graph connected?
 - Used to optimize roads or network setups
 - "Do I really need a cable from Math/Sci to Cosmo?"
- Slight change to how we'll use our graph tech - we'll consider our edges bidirectional instead of having arrows

Prim's Algorithm

- There are a few different algos to do this. Yes you have to use Prim here.
- Start with all vertices disconnected
 - Internal: Each node gets mConnected next to mIns and mOuts
 - External: Two lists for Connected and Unconnected
 - Copy: Literally remove all edges
- Start with any vertex and mark it Done
- Keep finding the smallest weight edge that connects *any* Done node to *any* not Done node

Greedy Again

- All of these algos are deceptively simple because they are greedy
- Always pick a zero vert, always pick the lowest number, none of these ever think ahead
- Most importantly, none of them consider the entire graph, so none of them fork, so none are recursive
 - Don't think recursive just because there are nodes and arrows



End

We can talk about proofs if you want, but people usually want more lab time