CS280 – Even More Graphs
March 28, 2016

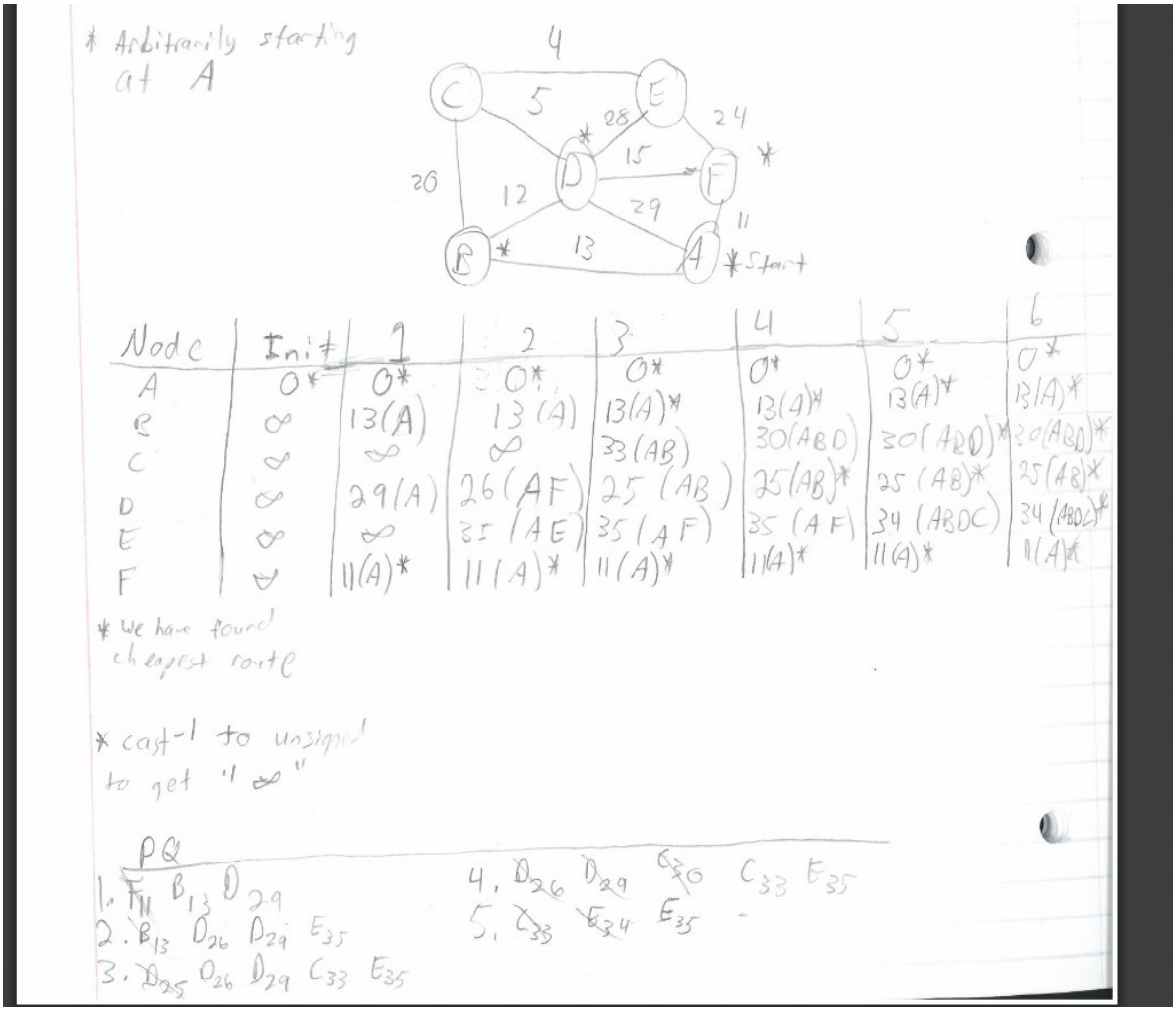
http://azrael.digipen.edu/~mmead/www/Courses/CS280/Graphs-1.html

Homework Tips

- Make sure and use Count + 1 when checking load factor, since we haven't grown table

Graphs

- We can merge the second step of finding the shortest path into the loop



- We are expected to make our own classes
    - Node
    - List
- Overload *operator*< for our priority queue to make life easy
- We might have directed and undirected graphs, nodes that have no neighbors, etc...
- If we have a tie, go to lower node value
- Assignment due on SUNDAY!
- *AddUEdge* will just call *AddDEdge*
- GraphViz is part of Dpxygen you can install
    - Open source to draw all sorts of data structures
- We will need to derive myStack/myQueu so that interface will work for both types of graphs
- The algorithm doesn't change for directed/undirected since we recieve the same data structure. Just a convience for the driver
- Start on assignment today
- Algorithm depends on underlying data structure
- Priority queu is implemented on the heap
- We can either sort as we build the graph and then run Dijkstra's, or just push back graph, do one big sort, then run Dijkstra's
- Using pipes in code to send output to console window
    - This is how visual studio sends compiler output to the console