March 22, 2019
CS: DS&A
Proof School

Assignment 24: Our First Shortest Path Algorithms Due in class Monday, 3/25

Problem 1. I'd like you to get very comfortable with the argument presented in class that our "brute-force" relaxation argument works, meaning that the $d(v)$'s it produces are the actual $D(v)$ minimum distances, and that it stops after $n-1$ iterations of relaxations.

Read through the write-up in the course notes. (This is the section called "Algorithm 2: Weighted Directed Graphs".) You have two concrete tasks:

1. I'd like everyone to come prepared to explain the argument to the class. (The "Verification" section.) I may call on someone to do this.
2. There's one conceptual error in the class notes in this "Algorithm 2: Weighted Directed Graphs" section. (There may be typos too, but those are not intentional.) Can you find it?

Problem 2. Let's try to combine relaxing with a Prim-like strategy of picking undiscovered vertices based on smallest weights.

Start with A, and first relax along edges coming out of A. Then, pick an unprocessed vertex with the smallest current d-value. Call that node processed, and relax along edges coming out of it. Keep going...

Analyze this algorithm. In particular, answer the the following questions.

a) How long do we have to do this? Is it enough to just go through all the nodes like this once, or do we have to do more than one iteration through all the nodes before the d-values are the same as the true D-values?

b) Give as careful a proof as you can that your answer to (a) actually works.

c) What can you say about the running time of this algorithm? How low can you get it? What data structure should you use to pick the nodes with minimum d-value effectively? Do you see any problems with using this data structure here?