CS 161: Section 6

Midterm Recap

1. **General midterm review.**

   See midterm solutions and/or Piazza; in section we did a midterm Q&A.

2. **How to do a formal proof by induction.**

   One of the things many people struggled with on the midterm was writing a formal proof by induction. Please take a look over your answer to problem 5, parts (c) and (d), at the TA's feedback, and at the solutions.

   Here is a list of resources to help you out with proving things by induction:

   CS103 Guide to Induction

   `http://web.stanford.edu/class/cs103/handouts/240%20Guide%20to%20Induction.pdf`

   CS103 Induction Checklist

   `http://web.stanford.edu/class/cs103/handouts/300%20Induction%20Proofwriting%20Checklist.pdf`

Dijkstra's Algorithm

In section we went through an example in detail; a great way to get this experience for yourself is to play around with this awesome website: `https://www.cs.usfca.edu/~galles/visualization/Dijkstra.html`

We also recapped the proof of Dijskstra's algorithm, so take a look at that in the course notes! It's a great example of a complex inductive argument.

Bonus Graph Problem: Farmville!

Suppose there is a set of farms in the central valley that have entered into a cooperative agreement. They built a series of irrigation pipes connecting the farms, with each pipe having a direction of flow (i.e. water can only travel in one, fixed, direction in each pipe). Each farm uses some of the water that passes through the pipes that go through the farm, and each farm also has a well that can contribute water to any irrigation pipes leaving the farm. [Think of this as a way of distributing the risk that certain wells might go dry one year, and others go dry a different year...some years Farmer Joe takes more water from the network than he contributes, some years he supplies more water from his well than he takes.]

1. How would you use graphs to model this problem?

2. After designing the network of (directed) pipes, the farmers want to ensure that water from every farm can get to every other farm. (For example, if there is only one farm whose well is still working, every farmer should still be able to get some of that water via the irrigation pipe network.) The only

algorithm that the farmers know is BFS, which takes as input a directed graph, $G = (V, E)$ and a source node $s \in V$, and runs in time $O(|V| + |E|)$ and outputs the set of nodes that can be reached from node s. Design a way for the farmers to check whether the desired condition holds for a given proposed pipe network, that uses their BFS algorithm only a *constant* number of times.

3. Thanks to your solution, the farmers built a successful network, and enjoy many years of steady crops. One year, however, a turnip gets stuck in an irrigation pipe, clogging that pipe; in the week it takes to fix that pipe, Farmer Joe's potatoes all perish from lack of water. The farmer coop forms a new emergency planning committee to design a new, improved irrigation plan. This time, they want a network such that the guarantees of the above part (namely that water from any farm's well can flow to any other farm via the irrigation pipes) will continue to hold even if any single pipe becomes clogged. Describe an algorithm for this part that uses your algorithm from the previous part, and give a bound on the runtime of the algorithm. (Justify the correctness of the algorithm, and the claimed runtime.)

4. What is the minimum number of irrigation pipes, as a function of the number of farms, n, that are necessary in any network that satisfies the desired property described in part (b)? Prove your answer– both that the claimed number can be achieved, and that no smaller number is possible.