Problem Set 6 - Network Flow

Problem E

Oh my goodness, the Evacuation implementation was absolute hell. The actual algorithm, not so much, but the implementation was so spicy I could probably write for hours about it. Reading the problem again, I also remember how immense the spec was, it was a whole essay lol.

This question felt great when initially trying to formulate an approach, felt very flow-y, we were just trying to get as many scientists into the capsules as possible - with my then accrued experience with flow, I could quickly identify that I wanted each of the capsule locations to be vertices on the flow network, and that I wanted each soldier to flow into them.

Then, adding in the broken nuclear reactor, it's pretty intuitive that we were trying to invalidate some of the edges going from the soldier to the eventual capsules. Initially I had thought that it had something to do with the way that Dinic runs flow, and the level traversal of the graph - I had the idea of, running Dinic from the reactor, and labelling each location with what level it was on the reactor graph (in hindsight, could definitely just be done with BFS).

Then, when flowing from the soldiers to each of the final locations, editing the dinic algorithm in some way to check if the level of the soldier flowing through is less than the level of the reactor flowing through, and then invalidating the flow that way.. In hindsight again, this approach doesn't really make too much sense, but I made sense at the time I was creating it.

A tutor then gave the insight that we probably shouldn't really be looking to edit the source code of the dinic algorithm, and the other black-boxes that we were given, so the better thing to do would be to check if each soldier could make it to any given capsule first, and then just use that information to determine whether to draw an edge from a scientist to a capsule. In this way, we just have a bipartite graph, rather than a full flow network showing all the paths through the lab.

To outline the solution, we have a source going to every scientist location, with an edge set to the amount of scientists in that location. These vertices would then be linked to the capsules that CAN be reached (shortest path < nuke's shortest path to that capsule). Then the capsules would be linked to the sink. This required me to keep a graph containing the possible paths through the entire laboratory, and bfs twice for each scientist/capsule pair to check if the scientist can reach the capsule.

Finally, running through the graph, we can find how many soldiers make it out to the capsules.