Problem Set 5 - Shortest Paths

Problem E

My initial instinct to try to solve President's Path, of course, was to use Floyd Warshall, but to try and augment it in some way to try and get ALL shortest paths for all pairs. And then do an iteration over each pair, traversing these shortest paths and summing up the edges used. This was a while ago, and pre the insight I got that we shouldn't really be messing with the black boxes in the lecture code provided us.

A few weeks later I returned to the question and, armed with this new knowledge, had a feeling I needed to do something with the output of the Floyd Warshall algorithm. Floyd Warshall gives you the length of the shortest path between each of the Pairs.

If we were to, for every pair of nodes (u, v), check EVERY intermediary i - similar to the normal loops of Floyd Warshall, but this time, check if the intermediary is the penultimate node on a shortest path between (u to v). To do so, we just check if:

```
dist[u][v] == dist[u][i] + edges[i][v]
```

If so, we'll add 1 the sum of the number of penultimate shortest path edges going into each node v from a source u. Finally - pretty elegantly so - if we then run the Floyd Warshall loops a third time, we can check if every intermediary is ON the shortest path from the current u to v, and if so, add the number of ways to get to that intermediary that takes a shortest path, which was already pre-computed from the prior loop. Sum all of these together for every intermediary, and we have the number of roads from every v to every u.

The implementation for this solution was incredibly clean, and probably the easiest of all of the E's I've done. E's are so cursed…