Problem Set 5 - Shortest Paths

<u>Problem D</u>

Completing the graph was a bit of a spicier one in terms of coming up with the approach, definitely took more time than the other questions in problem set 5. Since we're looking at shortest paths from a single source, with no negative edge weights, I realised that we were working with Djikstra. Exactly what to do with Djikstra? I wasn't exactly too sure at the beginning.

My first attempt at formulating an approach was to find the shortest path from the source to the destination - and if there were multiple, I made the observation that we always wanted to choose the one with the least amount of 0's. Since we were assigning positive integer weights, if we could augment the shortest path to length L with (n+1) 0's, then we could do it with (n) 0's.

Then, I would simply assign the 0's by looping through this path, assigning with 1's the whole way through, until the last 0, in which you would assign with the remaining units to be assigned from the difference of L and the shortest path. The main issue with this (among many) is that it doesn't have visibility over the larger graph. One of my friends showed me a case where the assignment order matters, and assigning in a certain way would make it such that the current path is no longer the shortest path. Since I was just arbitrarily assigning weights, my solution wouldn't work.

The correct solution I came to was to first, do Djikstra's from s to t, stopping where we encounter 0 length edges. After this, we would then run Djikstra's from t to s. This time, if we encounter a 0 length edge (let's say from u to v), we would do the edge allocation here at this step. Given the Djikstra from t is currently at u, if S' Djikstra reached v (dist != infinity), we would allocate it (L - ds[v] - dt[u]). If S' Djikstra hasn't reached v, there are multiple more 0's following on this path, and thus, we allocate it 1 instead.

Finally, once we've done all of these allocations, and we've reached S, we now know FOR SURE what our shortest path between s and t is, and if it is equal to L post our allocations, then we have a solution.