PRINCETON COS CON

Problem 10: Drop the Bass II

By Henry Tang

We want to find the minimum number of vertices that can be removed from the graph in order to stop all paths from 1 to n. We can solve problem by treating it as a maxflow problem. The idea is to create a new graph G' whose edges (each of capacity 1) will correspond to the vertices of the original graph G. This is because we can only perform maxflow on edges, not vertices! Transform each node v in G into two nodes labelled $2v-1$ and $2v$. For all inedges to v in G, now change them into inedges to $2v-1$ in G'. For all outedges from v in G, now change them to outedges from $2v$ in G'. Also attach a directed edge from $2v-1$ to $2v$. If we treat each edge as having capacity 1, then the maxflow in G' from node 2 to $2n-1$ is our desired answer.

The reason for this is that removing a vertex v in G is equivalent to filling the edge $2v-1$ and $2v$ in G' with full flow, because it represents invalidating all the inedges and outedges of vertex v in G.

However, most maxflow algorithms are fairly slow and have a time complexity polynomial in m and n. Luckily, the graph we have constructed is special. In particular, it is a simple unit-capacity network. A unit capacity network is a flow network where

- Every edge has capacity 1
- Every vertex has exactly 1 inedge, 1 outedge, or both

In such a scenario, running Dinic's algorithm (which typically takes $\mathcal{O}(n^2m)$ time, now runs in $\mathcal{O}(m\sqrt{n})$) time. This is sufficient to pass the test cases in this problem.

Time Complexity: $\mathcal{O}(m\sqrt{n})$, with the algorithm's runtime being dominated by the maxflow algorithm.

Plaudits:

- Congratulations to Antonio Molina (grad) for being the first to solve this problem at 114 minutes!