CS 6505 - Homework 7

Caitlin Beecham (Worked with)

4

We use something similar to Dijkstra's Algorithm. The idea is to maintain the minimum capacity among edges along the path to each vertex such that that minimum capacity is maximized among all such paths to that vertex. We also keep a Hashmap of each vertex to the previous vertex on this desired path we are constructing.

```
1. $C(s) = \infty$
2. $C(u) = -\infty$
3. Prev(s) = s.
4. Prev(u) = null for all $u \neq s$.
5. $S = \{s\}$
6. While $S \neq V$:
    (a) Let j be a vertex not in S.
    (b) For all vertices $v \in S$ such that there is an edge from v to j:
        i. If $min\{C(v), c_{vj}\} > C(j)$: $C(j) = min\{C(v), c_{vj}\}$ and update Prev(j) = v
        ii. else, pass.
    (c) $S = S \cup \{j\}$
```

At the end of this algorithm, the result of reading Prev(t), Prev(Prev(t)), Prev(Prev(Prev(t))), ... s gives the desired path (though in reverse order so we reverse to get the exact answer).