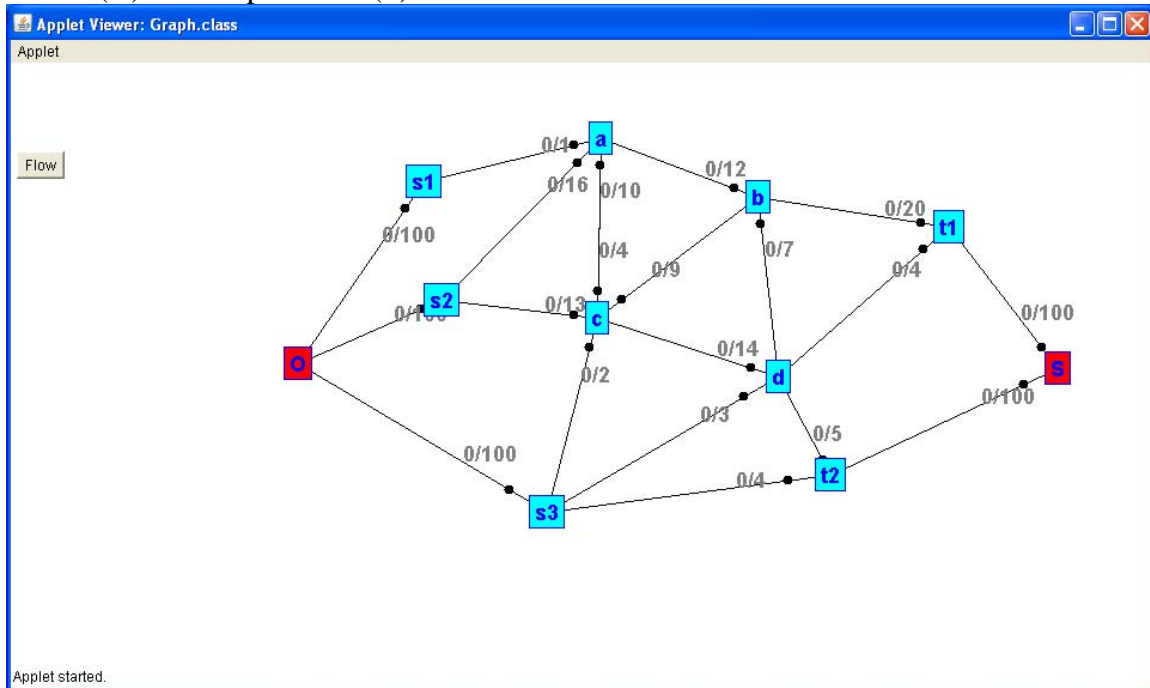


CS 5050 - Program #3 - 20 points

Network Flow - Edmonds-Karp Algorithm

Using your own graph code or pull the code from the class homepage (which has been adapted for a max flow problem), implement a flow graph. This was allowed in the original code. The format of the edges is **from-to/flow**. Implement the Edmonds-Karp network flow algorithm for the three graphs supplied with the code (in the edge String). In the graph below, note there are two edges between a and c. (You might want to try it with a single edge at first.) Note, too, that originally there were multiple sources (s1,s2,s3) and multiple sinks (t1,t2), but that this has been dealt with by adding a “super” source (O) and “super” sink (S).



One problem with Ford-Fulkerson is that running time can be large if capacities are large. The number of iterations is $O(F)$, where F is the max flow. This might not be polynomial in the description length of the problem instance. This is a problem for us, as complexity is normally expressed in terms of the size of the problem (number of arcs, nodes, etc.) not in terms of some extra number. (When a polynomial time algorithm has complexity that depends on an extra number, we call it pseudo-polynomial.) Thus, it would be nice if the running time were only dependent on the problem size.

Edmonds-Karp gives two natural heuristics that solve the problem.

- Heuristic 1: Always pick the maximum bottleneck augmenting path. (The path of maximum minimum capacity)
- Heuristic 2: Always pick the path having the fewest edges.

For this assignment, you are to implement heuristic 2. The following should be present in your user interface.

- 1) Allow the user to select from two possible views of the graph: flow graph and residual graph.
 - a) The flow graph contains nodes and arcs annotated with the total capacity, current flow, and network flow.
 - b) The residual graph will be a directed graph showing the unused capacity of forward edges and flow on backward edges.
- 2) Allow the user to select two modes of operation.
 - a) Automatic mode runs the algorithm and shows the final result.
 - b) Step-by-step mode shows each augmenting path that is found by high-lighting the path on the residual graph, and requires a button click to proceed to the next step.

Hint: Because (during the course of the algorithm) you need to have arcs going in each direction, I would initially create arcs in both directions – and just modify the flow on them. So, initially, the maxflow on the reverse edges is zero.

ReadMe File

1. How many hours did this assignment take?
2. Did everything work correctly? If not, what problems are there in your code?
3. Any thing the grader should know?