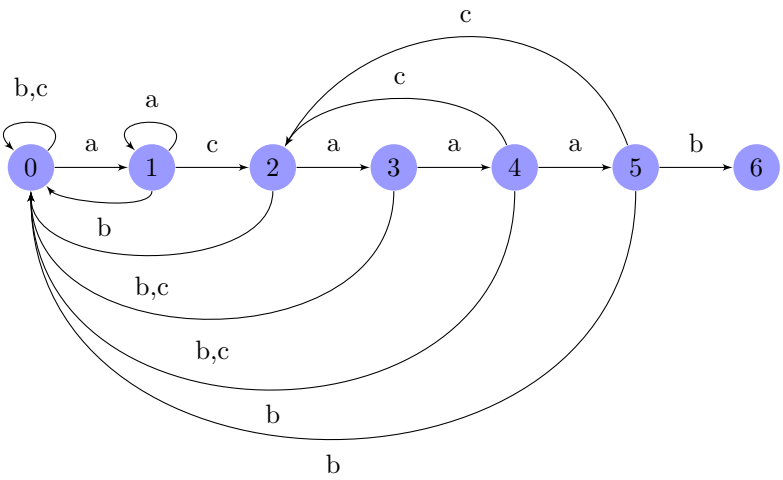
CSC 226 - Assignment 4 - Theory

Daniel Frankcom

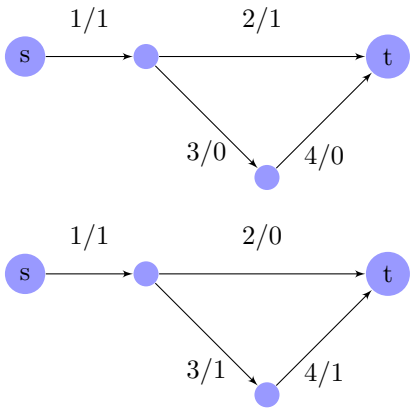
April 2017

1.



2. Any example where both the search and pattern strings are entirely made up of one character performs poorly, as the algorithm is not able to skip forward when comparing characters.
   Search string: "AAAAAAAAAAAAAAA"
   Pattern string: "AA"

3. (a) This statement is false. A counterexample for this is the following:

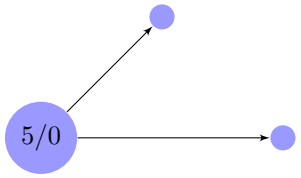


   (b) This is true. For any given maximum flow with a positive flow cycle, we can reduce the flow of each edge in the cycle so that at least one of the edges has flow 0, while maintaining the overall flow of the graph.

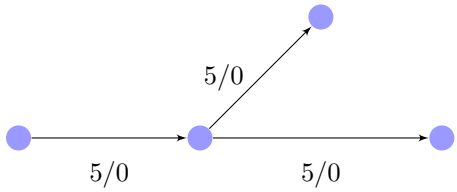
(c) This is true. Since all edge capacities are increased by the same amount, the hierarchy of edges remains the same. Since the algorithm to find the min cut inspects the relative order of the edges, if this remains unchanged, then the min cut will also remain unchanged.

4. This problem is tough, as we cannot simply apply the node capacity to each of the outgoing edges, as multiple edges would cause the node to gain more overall output capacity.

   An example of a problem graph is the following:



   We can deal with this issue by adding more nodes to the graph. Specifically, we will add a node to accept incoming edges, which will then point to our original node.



   In this manner, the total flow through the original node will be limited by the incoming node. The outgoing edges from our original node will have capacity c_v. This will ensure that if all of the flow needs to go through a single edge, it still can.

5. This problem can be simply reduced to a max flow problem by creating a residual graph where each edge in G is represented as an edge with flow 1. In this manner, when an arbitrary path is chosen during the computation of the max flow, the 1 unit of flow will no longer be available, so we will not choose another path which repeats the same edge.