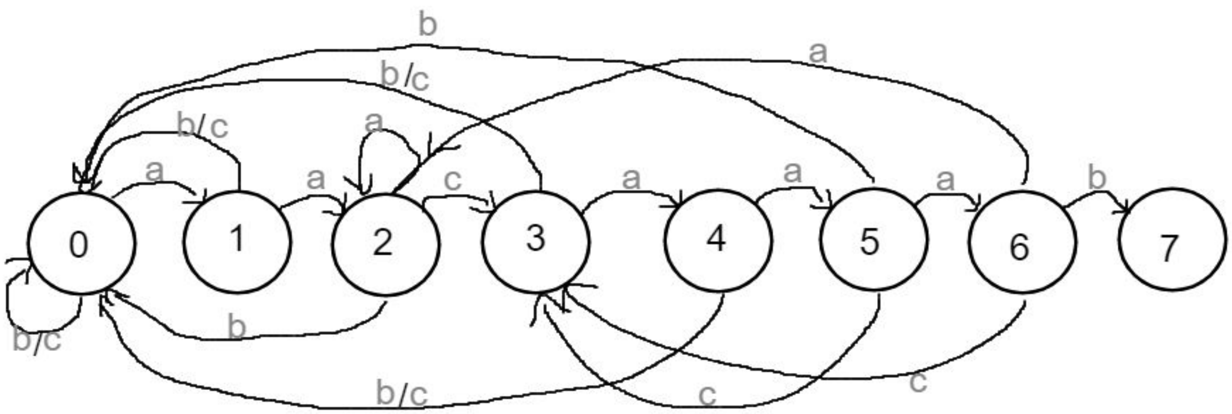
Nigel Decontie
V00853112

CSC 226 Assignment 4 Written

1. **Draw the KMP DFA for the following pattern string: AACAAAB.**

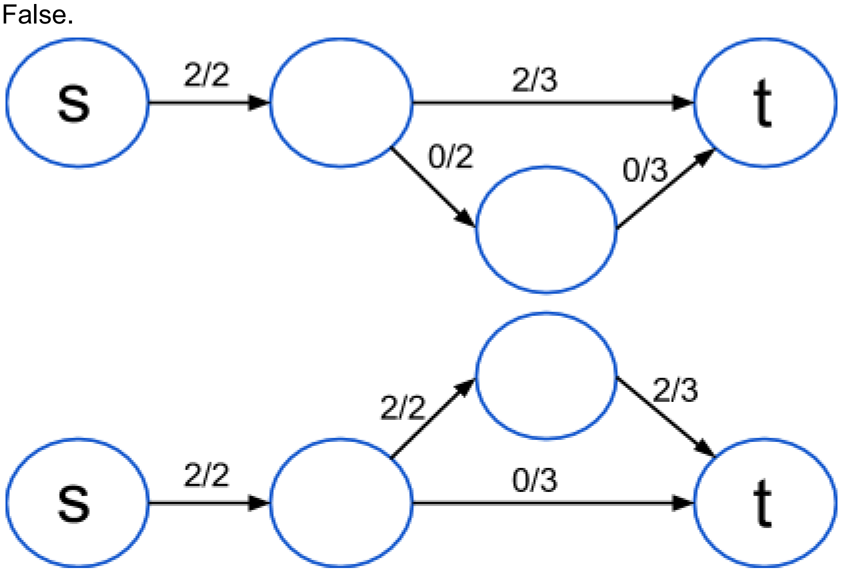


2. **Construct an example where the Boyer-Moore algorithm performs poorly:**

T = aaaaaaab
P = ab
The majority of the skips only shift by one element, giving the maximum run time of O(nm).

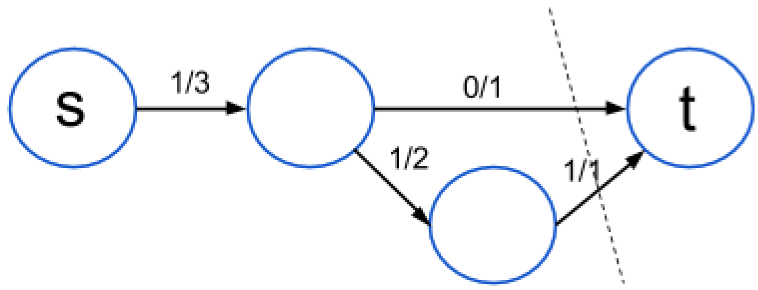
3. **True or false. If true, provide a short proof. If false, give a counterexample.**

   a. **If all edge capacities are distinct, the max flow is unique.**
   False.

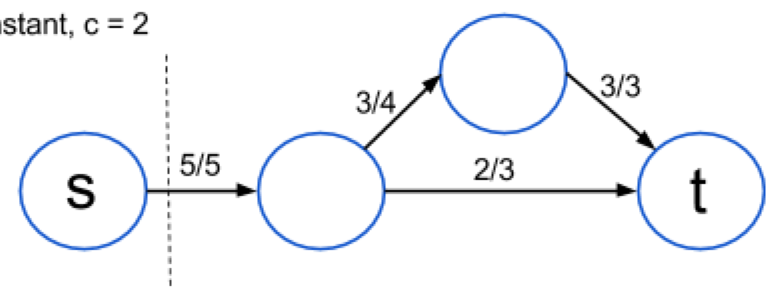


b. **There exists a max flow for which there is no directed cycle on which every edge carries positive flow.**
True.
Let ***f*** be the maximum flow.
Let ***C*** be a cycle with positive flow.
Let ***q*** = min [e∈C] f(e). Reducing each edge's flow in ***C*** by ***q*** keeps the flow's value and sets the flow f(e) of at least one of the edges e ∈ ***C*** to zero.

c. **If all edge capacities are increased by an additive constant, the min cut remains unchanged.**
False.



Constant, c = 2



4. **Show how Ford-Fulkerson algorithm can be used to find a maximum flow in a node capacitated network.**

The Ford-Fulkerson algorithm works by finding an augmenting path and pushing flow through the edges. It repeats this process until it cannot find another augmenting path, meaning there is no capacity for flow either by:

a. No non-full forward edges.
b. No non-empty backwards edges.

By modifying the algorithm to check for non-full nodes, we don't have to check for different kinds of edges and simply find nodes that have not reached full capacity. While paths exist, flow can be pushed through until no more paths exist (ie. there are no routes from s to t that can have more flow pushed through).

5. **Two paths in a graph are edge-disjoint if they have no edge in common. Disjoints Path Problem: Given a digraph G = (V, E) and two nodes s and t, find the maximum number of edge-disjoint s-t paths. Sow how this problem can be reduced to the max flow problem.**

In the max flow problem, the maximum number of edge-disjoint paths must be found in order to check all possible augmenting paths. To find a number *k* edge-disjoint paths, we can set the capacity of each edge to 1 and push one unit of flow along each path *s* to *t*. The value of the resulting flow is exactly *k*.