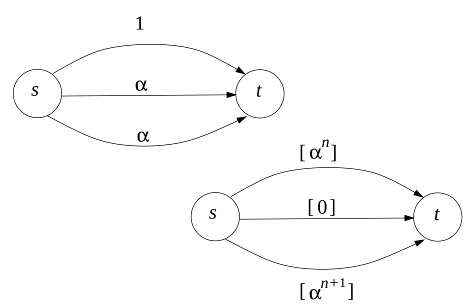
MA346m Quiz 05, due noon, Friday 15/12/17

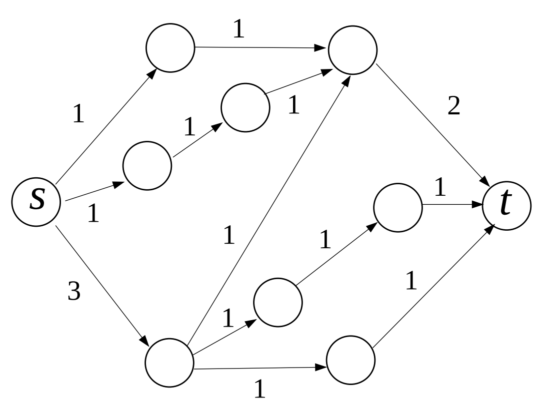
Answer 2 questions, as usual

(1). The Ford-Fulkerson algorithm may run forever. The figure below illustrates the idea. Admittedly it is on a multigraph; but it's easy to convert to a flow network in the usual sense. The figures in square brackets are the residual capacities, though sometimes in a forward and sometimes in a backward direction. Of course, α must satisfy a certain quadratic equation.



Calculate α and exhibit an infinite sequence of augmentations.

(2). Calculate a maximal flow, on the network shown below, by several augmentations. Your answer should include a minimum-capacity cutset.



(3). There is a curious application of network flows. Given a list $(i_1, o_1), \ldots, (i_n, o_n)$ of pairs of nonnegative integers, to construct a digraph G whose j-th vertex has indegree i_j and outdegree o_j, if it exists, and to show that no such digraph exists, otherwise.

Describe how to convert the 'degree problem' to a network flow problem.

(4). There is a bijection between binary trees on n nodes with balanced sets of $2n$ parentheses, allowing one alternative ways to count both. It is also the number of ways the numbers $1 \ldots n$ can be output by pushing them on a stack with intermittent popping.

For each n, the number of trees/balanced sets/restricted permutations is the n-th Catalan number

$$\frac{1}{n+1}\binom{2n}{n}$$

Derive this estimate for any one of the three structures mentioned above.

This can be calculated in at least three ways: by using the general Binomial Theorem on a generating function, by Lagrange inversion on same, or by a trick mentioned in 'introduction to probability theory…' by Feller.

To save time, here is a derivation of the generating function. Let b_n be the number of binary trees on n nodes and let $B(x) = \sum_n b_n x^n$. We take it that $b_0 = 1$.

A tree with $n+1$ nodes separates into left and right subtrees plus the root. If the left and right subtrees have i and j respectively then $i + j = n$. Therefore

$$\begin{aligned} b_{n+1} &= \sum_{i+j=n} b_i b_j \\ b_{n+1}x^{n+1} &= x \sum_{i+j=n} b_i x^i b_j x^j \\ B(x) - 1 &= xB^2(x). \end{aligned}$$

(5). True or false: Tarjan's SCC algorithm emits sccs according to the postorder rank of their roots. (Explain your answer.)