Graph Algorithms

Robin Whitty at www.theoremoftheday.org

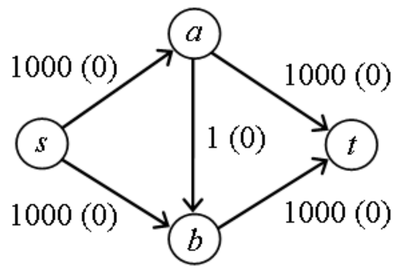
Question set for Chapter 6

1. Prove the following lemma, which is an important detail in a complete proof of correctness of Ford-Fulkerson:

   Lemma Suppose f is an st-flow in a network N and P is an f-unsaturated st-path in N. Suppose that flow g is constructed from f by increasing the value by the slack value $\mathrm{sl}((P))$ on forward arcs of P and decreasing the value by $\mathrm{sl}((P))$ on backward arcs of P. Then g is an st-flow with value $|g| = |f| + \mathrm{sl}((P))$.

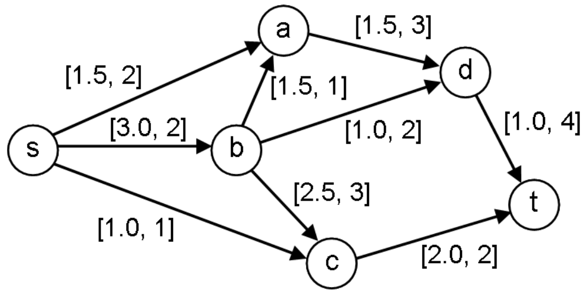
2. The network shown below right is from our recommended further reading: the book by Papadimitiou and Steiglitz.

   Starting from the initial zero-value flow (shown in brackets), run the Ford-Fulkerson algorithm for two iterations, using the **longest** possible flow augmenting path in each iteration. Hence predict the number of iterations which will be required, in the **worst case**, to maximise the flow. What conclusion to you draw about the complexity of the Ford-Fulkerson algorithm?



3. In the network below right each edge e is labelled with a pair of weights $[u(e), c(e)]$. The first entry is the *cost* of using the edge, the second entry is the *capacity* of the edge.

   Use trial and error to find an st-flow f which maximises $|f|$, subject to the edge capacities, while minimising cost, where the cost of a flow is the sum of the costs of the edges with non-zero flow, i.e. cost$=\sum_{e.f(e)\neq 0} u(e)$. Apply the Ford-Fulkerson algorithm, starting with a zero-flow, to find a maximum st-flow. Does this minimise cost? If so, do you think Ford-Fulkerson is guaranteed to *always* minimise cost as well as maximising flow?



4. The network below right has a flow f from s to t with initial values of $f(e)$ given in brackets on each edge e, the unbracketed number being the capacity $c(e)$ of the edge. *There is a 1-page pdf file containing a large version of the network image available on the Chapter 6 web page: you may like to print this page and write your answers on in order to save you redrawing the network by hand when answering this question.*

   Run the Ford-Fulkerson Algorithm for **three** iterations on the network in order to find a maximum flow and a minimum capacity cut.

   You should construct a maximal f-unsaturated tree during each iteration and identify, if possible, a flow augmenting path in each tree. This path should then be used to increase the value of the flow; while if no path can be found then, instead, a minimum capacity cut should be identified.

   You may present your work in any way you wish provided that you make clear how the Ford-Fulkerson Algorithm is working during each iteration. *Some further advice on this is given on the accompanying 1-page image document.*