CSL 356: Analysis and Design of Algorithms **Date:** September 20, 2011
Instructor: Ragesh Jaiswal

7.1 Another algorithm for max-flow

Consider the following slightly changed version of the Ford-Fulkerson max-flow algorithm. This algorithm is also due to Jack Edmonds and Richard Karp.

Max-Flow
- Start with a flow f such that $\forall e \in E, f(e) = 0$.
- While there is an $s - t$ path in G_f
    - Find an $s - t$ path P in G_f with **largest bottleneck value**.
        - Augment along P to obtain f'.
        - Update f to f' and G_f to $G_{f'}$

return(f).

1. Think of an algorithm to find the *largest bottleneck path* from s to t in a given graph. A bottleneck path is a path such that the bottleneck edge has maximum weight. Discuss its running time.
(*Hint: Try ideas from Dijkstra's Algorithm.*)
2. Let f be any $s - t$ flow and t be the value of maximum flow in the residual graph G_f. Let f' be the new flow after one augmentation and t' be the value of the new maximum flow in the residual graph $G_{f'}$. Argue that $t' \leq (1 - 1/m) \cdot t$.
3. Use the properties you showed above to argue that for a graph with integer capacities, the algorithm runs in time $O(m^2 \cdot \log m \cdot \log f^*)$, where f^* is the value of the max-flow in the original graph G.

7.2 Perfect matching in regular bipartite graph

A bipartite graph $G = (V, E)$, where $V = L \cup R$, is called *d-regular* if every vertex $v \in V$ has degree exactly d.

1. Show that for every d-regular bipartite graph $|L| = |R|$.
2. Use ideas from max-flow to argue that every d-regular bipartite graph has a matching of size $|L|$.