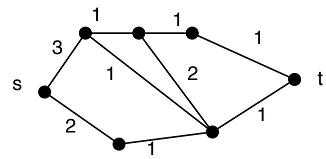
Graph theory - problem set 4

March 16, 2017

Exercises

1. Find a maximum flow from s to t and a minimum s-t cut in the following network.



2. What is the size of the maximum flow between two arbitrary vertices in the following networks?
   (a) C_n ($n \geq 3$) such that all edges have capacity 1.
   (b) K_n ($n \geq 2$) such that all edges have capacity 1.

3. Find a network on the vertex set $\{s, t, u, v\}$ with integer edge capacities such that the Ford-Fulkerson algorithm might make over a million improvements before reaching a maximum flow.

4. Prove Kőnig's theorem using the max flow-min cut theorem.

Problems

5. Let G be a network with source s, sink t, and integer capacities. Prove that an edge e is saturated (i.e., the flow uses its full capacity) in every maximum s-t flow if and only if decreasing the capacity of e by 1 would decrease the maximum capacity of an s-t flow in G.

6. The *Edmonds-Karp algorithm* is the same as Ford-Fulkerson, except it always chooses an augmenting path of shortest length (this can be found using BFS). In this problem we show that this small requirement can significantly improve the running time of the algorithm: no matter what the capacities are (large or irrational), it finds a max flow in polynomial time (in $O(|V||E|^2)$ steps, to be precise).
   (a) Let $l_a(v)$ denote the length of the shortest augmenting path from s to v after a steps. Show that if the algorithm chooses the augmenting s-t path $v_0 \ldots v_k$ after step a, then $l_a(v_{i+1}) = l_a(v_i) + 1$ for every $0 \leq i \leq k-1$.
   (b) Prove that these distances cannot decrease, i.e., $l_a(v) \geq l_{a-1}(v)$ for every a and v.
   [Hint: for fixed a, choose a v such that $l_a(v) < l_{a-1}(v)$ and $l_a(v)$ is minimum]
   (c) Show that if $l_i(t) = l_a(t)$ for some $i \geq a$, then after step i the algorithm saturates an edge uv such that $l_a(v) = l_i(v) = l_i(u) + 1 = l_a(u) + 1$ and does not unsaturate any other edge with this property. We get $l_{a+|E|}(t) > l_a(t)$ by noting that there are at most $|E|$ edges to saturate.
   [Hint: Look at the augmenting path. Use (b) on l_i and on the analogously defined $m_i(v)$ for the shortest length of an augmenting v-t path (i.e., that $m_a(v) \geq m_{a-1}(v)$)]
   (d) Deduce that the algorithm stops after at most $|V||E|$ improvements. We get a $O(|V||E|^2)$ running time using a $O(|E|)$-time BFS to find augmenting paths.