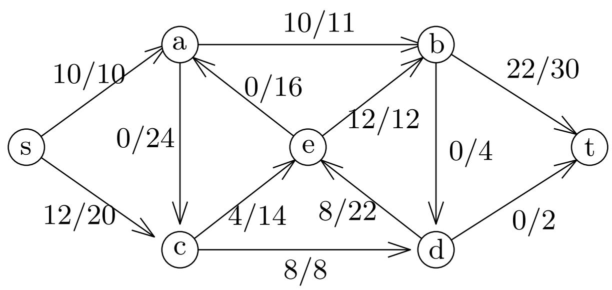
CS 473, Fall 2017
Homework 7 (due Nov 1 Wednesday at 8pm)

You may work in a group of at most 3 students. Carefully read `http://engr.course.illinois.edu/cs473/policies.html` and `http://engr.course.illinois.edu/cs473/integrity.html`. One member of each group should submit via Gradescope.

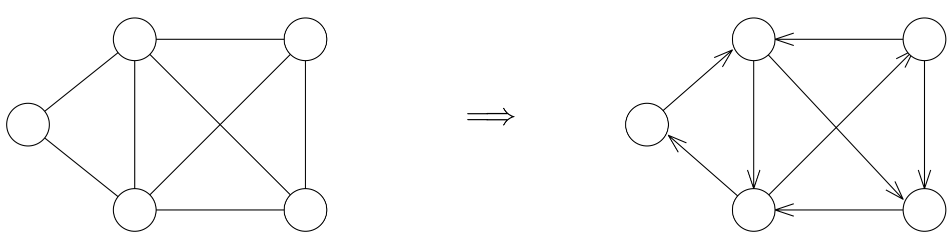
1. [*10 pts*] Run the Ford–Fulkerson algorithm for the maximum flow problem on the following example, starting with the initial flow shown. Here, the x/y label on an edge indicates that the edge has capacity y and initial flow value x. At each iteration, show the residual graph G_f, the augmenting path chosen, and the new flow values. Choose the augmenting path with the largest bottleneck value at each iteration.

   At the end, also show the minimum s-t cut generated by the algorithm.



2. [*13 pts*] Given an undirected graph $G = (V, E)$ and an integer d, we want determine whether it is possible to direct the edges so that the resulting directed graph has maximum **out-degree** at most d. Describe how to solve this problem by reduction to maximum flow. Prove correctness of your method.

   Example: for the undirected graph below (left) and $d = 2$, the answer is yes, and one solution is shown on the right (there are many other solutions).



   Hint: start with a bipartite graph where the vertices on the left side are the edges in G and the vertices on the right side are the vertices in G. Then add a source and a sink, set capacity of each edge appropriately…

3. [*27 pts*] Given a bipartite graph $G = (V, E)$ with n vertices and m edges, we want to find the largest *independent set* I, i.e., a subset $I \subseteq V$ such that no two vertices in I are adjacent in G.

One way to solve the problem is to construct a flow network (a directed graph) $G' = (V \cup \{s,t\}, E')$, where s is the source and t is the sink. Let V_L and V_R denote the left and right side of V in G. For each $u \in V_L$, we add the directed edge (s,u) to E'. For each $v \in V_R$, we add the directed edge (v,t) to E'. For each $uv \in E$ with $u \in V_L$ and $v \in V_R$, we add the directed edge (u,v) to G'. All edges have capacity 1. (This is the same flow network we have used to reduce maximum bipartite matching to maximum flow.)

(a) [*6 pts*] Prove that if there is an independent set in G of size k, then there is an (s,t)-cut in G' of capacity $n-k$.

(b) [*7 pts*] Conversely, prove that if there is an (s,t)-cut in G' of capacity $n-k$, then there is an independent set in G of size k.

(c) [*2 pts*] Conclude that there is a polynomial-time algorithm to compute a largest independent set in G.

(d) [*12 pts*] *A geometric application.* Given a set P of n points in 2D, we want to find a subset $Q \subseteq P$ of the largest size such that no two points in Q have Euclidean distance more than 1. (This problem is motivated by applications to clustering: we can think of Q as a cluster of points.) Describe a polynomial-time algorithm to solve this problem, using part (c) as a black box. (Note: You can still do part (d) without knowing the solution to (a)–(c). Also, there is no need to optimize the running time so long as it is polynomial.)

Hint: first pretend that we know the farthest pair q_0, q_1 of points in Q, which has distance $d(q_0, q_1) \leq 1$. The remaining points $R = Q - \{q_0, q_1\}$ must lie inside the intersection Z of two circular disks centered at q_0 and q_1 of radius $d(q_0, q_1)$. What does Z look like? We want a largest subset R in Z where no two points in R have distance more than $d(q_0, q_1)$. Why does the problem reduce to independent set in a *bipartite* graph?