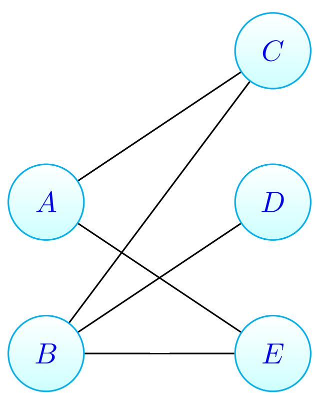
Name: Daniel Kim
ID: 102353420

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Quiz 8 Q1 S17 **Spring 2020, CU-Boulder**

Instructions: This quiz is open book and open note. You **may** post clarification questions to Piazza, with the understanding that you may not receive an answer in time and posting does count towards your time limit (30 min for 1x, 37.5 min for 1.5x, 45 min for 2x). Questions posted to Piazza **must be posted as PRIVATE QUESTIONS.** Other use of the internet, including searching for answers or posting to sites like Chegg, is strictly prohibited. Violations of these are grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

Standard 17. Let $G(V, E)$ be a graph. A *matching* of G is a set of edge $\mathcal{M}$ such that no two edges in $\mathcal{M}$ share a common vertex. That is, if $(i, j), (u, v) \in \mathcal{M}$ are distinct edges, then $i \neq u, i \neq v$, $j \neq u$, and $j \neq v$.

A graph is *bipartite* if its vertices can be partitioned into two sets $V(G) = L \cup R$ such that every edge has one endpoint in L and one endpoint in R. Note that L and R are disjoint. The graph pictured below is an example.



Consider the following problem

Bipartite Maximum Matching
Input: A bipartite graph $G = (L, R; E)$
Output: A matching $\mathcal{M} \subseteq E(G)$ whose size $|\mathcal{M}|$ is as large as possible.

(a) Describe how to reduce the above problem to the (one-source, one-sink) max-flow problem from class. Your description should be **general**, and not tied to a specific example. (You may illustrate with an example for expository purposes, but an example alone is not sufficient. E.g., "This is how my construction is performed in general. Then for example, this is how we apply the construction to the graph I selected.")

I would reduce the above problem, we need to modify the graph. We build a directed graph G' with L' containing all nodes from L and R' containing all nodes from R. So we need to add a source and you insert a directed edges from s to every source nodes in L'. Next, we need to add a new sink t with a directed edge added to every sink nodes in R'. We also give each edge a capacity of 1. So that we are letting source s bring as much flow for the multiple sources and letting sink t absorb as much flow for the multiple sinks. Since we set each edge with 1 capacity, the nodes in L has maximum of 1 outgoing and R has maximum of 1 incoming. By these conditions, we can use ford-fulkerson algorithm to get a maximum matching in G.

Name: Daniel Kim
ID: 102353420

(b) Using your reduction, find a maximum matching in the graph above. Show your work, as well as your final answer. Note that there may be multiple maximum matchings in the graph above; you need only find one such matching.

The first augmenting path is s to A to C to t. The bottleneck capacity of the path is 1 so the path is depleted after sending 1 unit of flow

The Second augmenting path is s to B to D to T, The bottleneck capacity of the path is also 1 so the path will be depleted after sending 1 unit of flow.

By augmenting path theorem, the maximum matching for this graph is 2.

Had no time to draw the example..