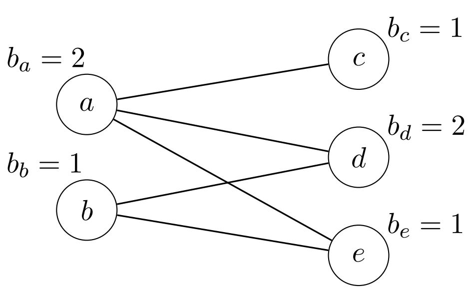
Advanced Algorithms — Exercise Set 3

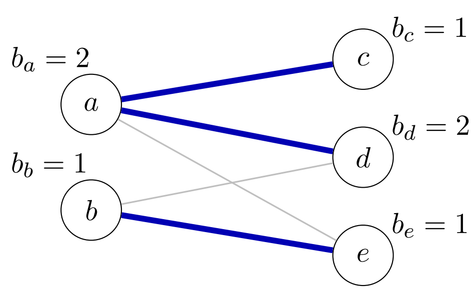
Name: ____________________________

- Submit on **Gradescope** by class time (1:15pm) on February 24.
- Feel free to discuss with others, but write up your own work.
- Half points on this exercise set are awarded for completion / effort. Use it to learn!

Problem 1. Suppose you are given an undirected bipartite graph $G = (L \cup R, E)$ and a positive integer b_v for every vertex v. A *b-matching* is a subset $M \subseteq E$ of edges such that each vertex v is incident to at most b_v edges of M. (The standard bipartite matching problem corresponds to the case where $b_v = 1$ for every $v \in L \cup R$.) See the picture below for an example of a b-matching instance, and a maximum-sized feasible solution shown.

Show how the problem of computing a maximum-cardinality bipartite b-matching reduces to the problem of computing a (standard) maximum-cardinality bipartite matching in a bigger graph.





Solution.

Problem 2. We have seen an algorithm for the maximum-size matching problem in a bipartite graph (via a reduction to max-flow). We have also seen an algorithm for minimum-cost bipartite perfect matching problem (via cycle-toggling). Think about how each of these algorithms relies on the bipartiteness of the input graph. Why would they not work for solving the corresponding problem in a general (non-bipartite) graph?

Solution.