Problem 1. Given is a network flow instance. A min-capacity of a path \((i_0, i_1, \cdots, i_k)\) is \(\min_{j=1}^{k} u_{i_{j-1}i_j}\). Describe an efficient algorithm, similar to Dijkstra’s that finds a path from \(r\) to \(s\) with the maximum min-capacity.

You can use a pseudocode from Lecture 30, but you need three changes: (1) The algorithm should return "NO PATH" if there is no path from \(r\) to \(s\), (2) it should return min-capacity and a list of nodes on such a path if it does exists, (3) of course, a small change is needed so you compute a min-capacity path rather than a shortest path.

Problem 2. Show an example of a network flow instance such that after
(a) finding an augmenting path from \(r\) to \(s\) with the maximum min-capacity,
(b) changing the capacities according to the flow sent on that augmenting path,
(c) finding again an augmenting path from \(r\) to \(s\) with the maximum min-capacity,
the path found in (c) has a larger min-capacity than the path found in (a).

Problem 3. We consider a network flow instance and shortest augmenting paths, i.e. paths from \(r\) to \(s\) with positive min-capacity and the minimum number of edges, say \(k\).

a. Suppose that \((i, j)\) is an edge on a shortest augmenting path and \(u_{ji} = 0\). Show that even if we increase \(u_{ji}\) to a positive value, \((j, i)\) cannot belong to an augmenting path of length \(k\) or less.

b. Show how to find all edges that belong to shortest augmenting paths in time \(O(|E|)\).

c. Assume that all positive capacities are equal to 1. Show how in time \(O(|E|)\) one can find shortest augmenting paths \(P_1, P_2, \cdots\) such that after changing capacities according to the flow sent through these paths there is no augmenting path of length \(k\) or less. (Perhaps it would be simpler to say that in time \(O(|E|)\) you should find a maximal set of shortest augmenting paths.)

d. Use the result of c. to show that given an instance \((V, E)\) of a bipartite matching problem in time \(O(\sqrt{|V||E|})\) we can find a matching \(M\) such that the shortest augmenting path for \(M\) has at least \(\sqrt{|V|}\) edges.

e. If \(M^*\) is a maximum matching, and for matching \(M\) there are no augmenting paths with less that \(\sqrt{|V|}\) edges, then \(|M^*| - |M| < \sqrt{|V|}\). Use this implication, and the result of d. to show that a maximum matching in a bipartite graph can be found in time \(O(\sqrt{|V||E|})\).
**Problem 4.** Find a maximum matching and a minimum cover in the bipartite graph below.

**Problem 5.** In a tournament $n$ players, $1, \ldots, n$, play each other once, and there are no ties. Let $w_i$ be the number of wins of player $i$. Describe and analyze an algorithm that determines if a vector $(w_1, \ldots, w_n)$ is a possible outcome of such a tournament. The algorithm should run in time that is polynomial in $n$. 