Tutorial 8, Design and Analysis of Algorithms, 2019

1. Using the *Ford-Fulkerson Algorithm*, solve the *Bipartite Matching Problem* for the instance of the graph given in figure 1.

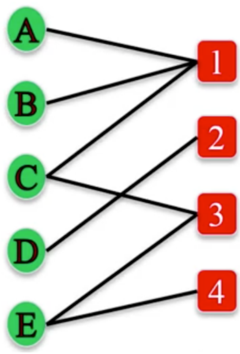

Figure 1: Figure for problem 1.

2. For the following directed graph, find the maximum number of edge disjoint $s-t$ paths using the Ford-Fulkerson's algorithm.

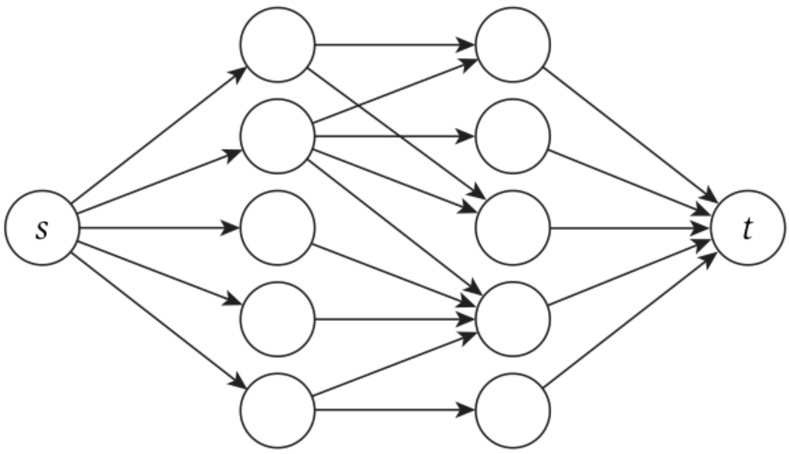

Figure 2: Graph for question 2.

3. In a standard $s-t$ Maximum-Flow Problem, we assume edges have capacities, and there is no limit on how much flow is allowed to pass through a node. In this problem, we consider the variant of the Maximum-Flow and Minimum-Cut problems with node capacities. Let $G = (V, E)$ be a directed graph, with source $s \in V$, sink $t \in V$, and nonnegative node capacities $\{c_v \geq 0\}$ for each $v \in V$. Given a flow f in this graph, the flow though a node v is defined as $f^{\text{in}}(v)$. We say that a flow is feasible if it satisfies the usual flow-conservation constraints and the node-capacity constraints: $f^{\text{in}}(v) \leq c_v$ for all nodes. Give a polynomial-time algorithm to find an $s-t$ maximum flow in such a node-capacitated network. Define an $s-t$ cut for node-capacitated networks, and show that the analogue of the Max-Flow Min-Cut Theorem holds true.

4. Give a polynomial-time algorithm for the following minimization analogue of the Maximum-Flow Problem. You are given a directed graph $G = (V, E)$, with a source $s \in V$ and sink $t \in V$, and numbers (capacities) $l(v, w)$ for each edge $(v, w) \in E$. We define a flow f, and the value of a flow, as usual, requiring that all nodes except s and t satisfy flow conservation. However, the given numbers are lower bounds on edge flow - that is, they require that $f(v, w) \geq l(v, w)$ for every edge $(v, w) \in E$, and there is no upper bound on flow values on edges.

   (a) Give a polynomial-time algorithm that finds a feasible flow of minimum possible value.

   (b) Prove an analogue of the Max-Flow Min-Cut Theorem for this problem (i.e., does min-flow = max-cut?).

5. Let LOOKUP denote the following function: on input a pair $< x, i >$ (where x is a binary string and i is a natural number), LOOKUP outputs the ith bit of x or 0 if $|x| < i$. Prove that LOOKUP $\in$ **P**.

6. Prove that the following languages/decision problems on graphs are in **P**: (You may pick either the adjacency matrix or adjacency list representation for graphs; it will not make a difference. Can you see why?)

   (a) CONNECTED the set of all connected graphs. That is, $G \in$ CONNECTED if every pair of vertices u, v in G are connected by a path.

   (b) TRIANGLEFREE the set of all graphs that do not contain a triangle (i.e., a triplet u, v, w of connected distinct vertices.

   (c) BIPARTITE the set of all bipartite graphs. That is, $G \in$ BIPARTITE if the vertices of G can be partitioned to two sets A, B such that all edges in G are from a vertex in A to a vertex in B (there is no edge between two members of A or two members of B).

   (d) TREE the set of all trees. A graph is a tree if it is connected and contains no cycles. Equivalently, a graph G is a tree if every two distinct vertices u, v in G are connected by exactly one simple path (a path is simple if it has no repeated vertices).

7. Recall that normally we assume that numbers are represented as string using the *binary* basis. That is, a number n is represented by the sequence $x_0, x_1, ..., x_{\log n}$ such that $n = \sum_{i=0}^{\log n} x_i 2^i$, where for each $i \in [0..\log n]$ $x_i \in \{0, 1\}$. However, we could have used other encoding schemes. If $n \in \mathbb{N}$ and $b \geq 2$, then the representation of n in base b, denoted by $[x]_b$ is obtained as follows: first represent n as a sequence of digits in $\{0, ..., b - 1\}$, and then replace each digit $d \in [0..b - 1]$ by its binary representation. The unary representation of n, denoted by $[n]_1$ is the string 1^n (i.e., a sequence of n ones).

   (a) Show that choosing a different base of representation will make no difference to the class **P**. That is, show that for every subset S of the natural numbers, if we define $L_S^b = \{[n]_b : n \in S\}$ then for every $b \geq 2$, $L_S^b \in \mathbf{P} \Leftrightarrow L_S^2 \in \mathbf{P}$.

   (b) Show that choosing the unary representation may make a difference by showing that the following language is in **P**:

   UNARYFACTORING $= \{< [n]_1, [l]_1, [k]_1 >$: there is a prime $j \in (l, k)$ dividing $n\}$

   It is not known to be in **P** if we choose the binary representation.