Problem Set 2

This problem set is not due and it will not be graded, but you are welcome to come to office hours to discuss your solutions. This is a set of practice problems for the midterm and final, and it covers the first quarter of the course.

The description of your proofs and algorithms should be as *clear* as possible (which does not mean *long* – in fact, typically, good clear explanations are also short.)

When a problem asks to give an algorithm, in your solution: (i) describe shortly and informally the main ideas in your solution; (ii) give a detailed description of the algorithm, using a style similar to the pseudo-code used in class or in a textbook; (iii) prove the correctness of the algorithm; (iv) prove a bound on the time complexity of the algorithm. You can omit the proof of correctness if it is clear from the description of the algorithm.

---

1. We are given a directed graph $G = (V, E)$ describing a network, and, for each edge $(u, v) \in E$ we are given the probability $p(u, v)$ that the link (u, v) may fail. These probabilities are independent, that is, the probability that no edge fails in the path $(u_1, u_2, \ldots, u_k)$ is $(1 - p(u_1, u_2))(1 - p(u_2, u_3)) \cdots (1 - p(u_{k-1}, u_k))$. Such a value is called the *reliability* of the path $(u_1, \ldots, u_k)$.

   Describe and analyse an algorithm that given G, the values $p(\cdot, \cdot)$ and two vertices $s, t \in V$, finds the path from s to t of maximum reliability.

   [Hint: if F is a monotone increasing function, then looking for the path $(s, u_1, \ldots, u_k, t)$ that maximizes $(1-p(s, u_1))(1-p(u_1, u_2)) \cdots (1-p(u_k, t))$ is the same as looking for the path that maximizes $F((1-p(s, u_1))(1-p(u_1, u_2)) \cdots (1-p(u_k, t)))$. Similarly, if F is monotone decreasing, the problem is the same as looking for the path $(s, u_1, \ldots, u_k, t)$ that minimizes $F((1 - p(s, u_1))(1 - p(u_1, u_2)) \cdots (1 - p(u_k, t)))$]

2. We have seen that, using the Ford-Fulkerson algorithm, not only we can find the flow of maximum cost, but also the cut of minimum capacity. In certain applications, it is important to know if the minimum cut is unique, and, if not, how many minimum cuts there are.

   (a) Give an example of a network where the minimum cut is unique.

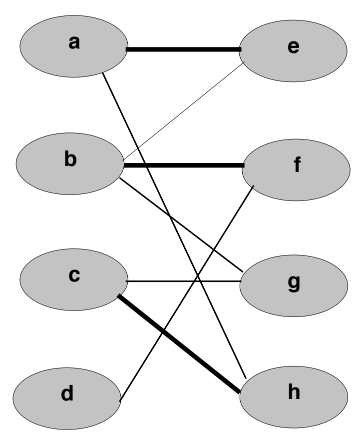
   (b) Give an example of a network where there is more than one minimum cut.

   (c) Give a polynomial time algorithm that determines whether the minimum cut is unique, and, if not, finds at least two distinct minimum cuts.

3. A cohort of k spies resident in a certain country needs escape routes in case of emergency. They will be travelling using the railway system which we can think of as a directed graph $G = (V, E)$ with V being the cities. Each spy i has a starting point $s_i \in V$ and needs to reach the consulate of a friendly nation; these consulates are in a known set of cities $T \subseteq V$. In order to move undeteced, the spies agree that at most c of them should ever pass through any one city. Our goal is to find a set of paths, one for each of the spies (or detect that the requirements cannot be met).

   Model this problem as a flow network. Specify the vertices, edges and capacities, and show that a maximum flow in your network can be transformed into an optimal solution for the original problem. You do not need to explain how to solve the max-flow instance itself.

4. Below we see a bipartite graph and a matching with 3 edges. Does the graph have a matching with 4 edges?



5. Find a maximum flow and a minimum cut in the network below