CMPSCI 311: Introduction to Algorithms **Spring 2018**

Homework 5

Released 04/04/2018 Due 11:59pm 04/18/2018 in Gradescope

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Instructions. You may work in groups, but you must individually write your solutions yourself. List your collaborators on your submission.

If you are asked to design an algorithm as part of a homework problem, please provide: (a) the pseudocode for the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm, and (e) justification for your running time analysis.

Submission instructions. This assignment is due by 11:59pm on 04/18/2018 in Gradescope. Please submit a pdf file. You may submit a scanned handwritten document, but a typed submission is preferred.

1. **Disaster Management, K&T Ch.7 Ex.9.** A natural disaster has hit the Pioneer Valley! The paramedics have n injured people that they need to get to k hospitals. Each person needs to get to a hospital within a half-hour driving time from their current location, so each person i has a set of hospitals they can go to $S_i \subset \{1, \ldots, k\}$. At the same time, the paramedics do not want to overload any hospital, so they are not allowed to send more than $\lceil n/k \rceil$ patients to any single hospital.

   Design an algorithm that the paramedics can use to decide whether it is possible to get injured people to the hospital, so that both the locality constraint and the overloading constraint are satisfied.

   Your algorithm need not output a assignment of people to hospitals, just whether one exists or not.

2. **Network Flows, K&T Ch.7 Ex.12.** You are given a flow network with unit-capacity edges: It consists of a directed graph $G = (V, E)$, a source $s \in V$, and a sink $t \in V$; and $c_e = 1$ for every $e \in E$. You are also given a parameter k. The goal is to delete k edges so as to reduce the maximum $s - t$ flow in G by as much as possible. In other words, you should find a set of edges $F \subset E$ so that $|F| = k$ and the maximum $s - t$ flow in $G = (V, E \setminus F)$ is as small as possible subject to this. Give a polynomial-time algorithm to solve this problem.

3. **Node Capacities, K&T Ch.7 Ex.13.** 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 a variant of the Maximum-Flow and Min-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 through 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. **Bipartite Graphs, K&T Ch.7 Ex.15.** Suppose you and your friend Alanis live, together with $n-2$ other people for a total of n, at a popular off-campus cooperative apartment, the Upson Collective. Over the next n nights, each of you is supposed to cook dinner for the co-op exactly once, so that someone cooks on each of the nights.

   Of course, everyone has scheduling conflicts with some of the nights (e.g., exams, concerts, etc.), so deciding who should cook on which night becomes a tricky task. For concreteness, lets label the people $\{p_1, \ldots, p_n\}$ and the nights $\{d_1, \ldots, d_n\}$. For person p_i, theres a set of nights $S_i \subset \{d_1, \ldots, d_n\}$ when they are not able to cook.

A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j, then $d_j \notin S_i$.

(a) Describe a bipartite graph G so that G has a perfect matching if and only if there is a feasible dinner schedule for the co-op.

(b) Your friend Alanis takes on the task of trying to construct a feasible dinner schedule. After great effort, she constructs what she claims is a feasible schedule and then heads off to class for the day. Unfortunately, when you look at the schedule she created, you notice a big problem. $n-2$ of the people at the co-op are assigned to different nights on which they are available: no problem there. But for the other two people, p_i and p_j, and the other two days, d_k and d_l you discover that she has accidentally assigned both p_i and p_j to cook on night d_k, and assigned no one to cook on night d_l You want to fix Alanis's mistake but without having to recompute everything from scratch. Show that its possible, using her almost correct schedule, to decide in only $O(n^2)$ time whether there exists a feasible dinner schedule for the co-op. (If one exists, you should also output it.)

5. **(0 points).** How long did it take you to complete this assignment?