Problem Set 8

Reading: Chapters 26.1–26.3

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered in the exercises.

Mark the top of each sheet with your name, the course number, the problem number, your recitation section, the date and the names of any students with whom you collaborated.

Three-hole punch your paper on submissions.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of the essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Full credit will be given only to correct algorithms that are *which are described clearly*. Convoluted and obtuse descriptions will receive low marks.

Exercise 8-1. Do Exercise 26.1-9 on page 650 of CLRS.

Exercise 8-2. Do Exercise 26.2-4 on page 664 of CLRS.

Exercise 8-3. Do Exercise 26.2-10 on page 664 of CLRS.

Exercise 8-4. Do Exercise 26.3-2 on page 668 of CLRS.

Problem 8-1. Inspirational fires

To foster a spirit of community and cut down on the cliqueishness of various houses, MIT has decided to sponsor community-building activities to bring together residents of different living groups. Specifically, they have started to sponsor official gatherings in which they will light copies of CLRS on fire.

Let G be the set of living groups at MIT, and for each $g \in G$, let $residents(g)$ denote the number of residents of living group g. President Hockfield has asked you to help her out with the beginning of her administration. She gives you a list of book-burning parties P that are scheduled for Friday night. For each party $p \in P$, you are given the number $size(p)$ of people who can fit into the site of party p.

The administration's goal is to issue party invitations to students so that no two students from the same living group receive invitations to the same book-burning party. Formally, they want to send invitations to as many students as possible while satisfying the following constraints:

- for all $g \in G$, no two residents of g are invited to the same party;
- for all $p \in P$, the number of people invited to p is at most $size(p)$.

(a) Formulate this problem as a linear-programming problem, much as we did for shortest paths. Any legal set of invitations should correspond to a feasible setting of the variables for your LP, and any feasible integer setting of the variables in your LP should correspond to a legal set of invitations. What objective function maximizes the number of students invited?

(b) Show how this problem can be solved using a maximum-flow algorithm. Your algorithm should return a set of legal invitations, if one exists, and return FAIL if none exists.

(c) (*Optional.*) Can this problem can be solved more efficiently than with a maximum-flow algorithm?

Problem 8-2. Zippity-doo-dah day

On Interstate 93 south of Boston, an ingenious device for controlling traffic has been installed. A lane of traffic can be switched so that during morning rush hour, traffic flows northward to Boston, and during evening rush hour, if flows southward away from Boston. The clever engineering behind this design is that the reversible lane is surrounded by movable barriers that can be "zipped" into place in two different positions.

For some reason, gazillions of people have decided to drive from Gillette Stadium in Foxboro, MA to Fenway Park. (They seem to be cursing a lot, or, at the very least, you hear them shouting the word "curse" over and over.) Governor Mitt asks you for assistance in making use of the zipper-lane technology to increase the flow of traffic from Foxboro to Fenway.

We can model this road network as directed graph $G = (V, E)$ with source s (Foxboro), sink t (Fenway), and integer capacities $c : E \to \mathbb{Z}^+$ on the edges. You are given a maximum flow f in the graph G representing the rate at which traffic can move between these two locations. In this question, you will explore how to increase the maximum flow using "zippered" edges in the graph.

Let $(u, v) \in E$ be a particular edge in G such that $f(u, v) > 0$ and $c(v, u) \geq 1$. That is, there is positive flow on this edge already, and there is positive capacity in the reverse direction. Suppose that zipper technology increases the capacity of the edge (u, v) by 1 while decreasing the capacity of its ***transpose*** edge (v, u) by 1. That is, the zipper moves 1 unit of capacity from (v, u) to (u, v).

(a) Give an $O(V+E)$-time algorithm to update the maximum flow in the modified graph.

Zap 86 years into the future! Zipper lanes are commonplace on many more roads in the Boston area, allowing one lane of traffic to be moved from one direction to the other. You are once again given the directed graph $G = (V, E)$ and integer capacities $c : E \to \mathbb{Z}^+$ on the edges. You also have a zipper function $z : E \to \{0, 1\}$ that tells whether an additional unit of capacity can be moved from (v, u) to (u, v). For each $(u, v) \in E$, if $z(u, v) = 1$, then you may now choose to move 1 unit of capacity from the transpose edge (v, u) to (u, v). (You may assume that if $z(u, v) = 1$, then the edge (v, u) exists and has capacity $c(v, u) \geq 1$. Again, you are given a source node $s \in V$, a sink node $t \in V$, and a maximum flow f. Governor Mitt IV asks you to configure all the zippered lanes so that the maximum flow from s to t in the configured graph is maximized.

(b) Describe an algorithm that employs a maximum-flow computation to determine the following:

1. the maximum amount that the flow can be increased in this graph after your chosen zippered lanes are opened; and
2. a configuration of zippered lanes that allows this flow to be achieved.

Because the graph G is actually a network of roads, it is nearly planar, and thus $|E| = O(V)$.

(c) Give an algorithm that runs in time $O(V^2)$ to solve the graph configuration problem under the assumption that $|E| = O(V)$. You should assume that the original flow f has already been computed and you are simply determining how best to increase the flow.