Problem Set 6

Reading: Chapters 22, 24, and 25.

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.

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, pseudo-code.
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 which are *which are described clearly*. Convoluted and obtuse descriptions will receive low marks.

Exercise 6-1. Do Exercise 22.2-5 on page 539 in CLRS.

Exercise 6-2. Do Exercise 22.4-3 on page 552 in CLRS.

Exercise 6-3. Do Exercise 22.5-7 on page 557 in CLRS.

Exercise 6-4. Do Exercise 24.1-3 on page 591 in CLRS.

Exercise 6-5. Do Exercise 24.3-2 on page 600 in CLRS.

Exercise 6-6. Do Exercise 24.4-8 on page 606 in CLRS.

Exercise 6-7. Do Exercise 25.2-6 on page 635 in CLRS.

Exercise 6-8. Do Exercise 25.3-5 on page 640 in CLRS.

Problem 6-1. Truckin'

Professor Almanac is consulting for a trucking company. Highways are modeled as a directed graph $G = (V, E)$ in which vertices represent cities and edges represent roads. The company is planning new routes from San Diego (vertex s) to Toledo (vertex t).

(a) It is very costly when a shipment is delayed en route. The company has calculated the probability $p(e) \in [0, 1]$ that a given road $e \in E$ will close without warning. Give an efficient algorithm for finding a route with the minimum probability of encountering a closed road. You should assume that all road closings are independent.

(b) Many highways are off-limits for trucks that weigh more than a given threshold. For a given highway $e \in E$, let $w(e) \in \mathbb{R}^+$ denote the weight limit and let $l(e) \in \mathbb{R}^+$ denote the highway's length. Give an efficient algorithm that calculates: 1) the heaviest truck that can be sent from s to t, and 2) the shortest path this truck can take.

(c) Consider a variant of (b) in which trucks must make strictly eastward progress with each city they visit. Adjust your algorithm to exploit this property and analyze the runtime.

Problem 6-2. Constructing Construction Schedules

Consider a set of n jobs to be completed during the construction of a new office building. For each $i \in \{1, 2, \dots, n\}$, a ***schedule*** assigns a time $x_i \geq 0$ for job i to be started. There are some constraints on the schedule:

1. For each $i, j \in \{1, 2, \dots, n\}$, we denote by $A[i, j] \in \mathbb{R}$ the ***minimum latency*** from the start of job i to the start of job j. For example, since it takes a day for concrete to dry, construction of the walls must begin at least one day after pouring the foundation. The constraint on the schedule is:

$$\forall\, i, j \in \{1, 2, \dots, n\} : \quad x_i + A[i, j] \quad \leq \quad x_j \tag{1}$$

   If there is no minimum latency between jobs i and j, then $A[i, j] = -\infty$.

2. For each $i, j \in \{1, 2, \dots, n\}$, we denote by $B[i, j] \in \mathbb{R}$ the ***maximum latency*** from the start of job i to the start of job j. For example, weatherproofing must be added no later than one week after an exterior wall is erected. The constraint on the schedule is:

$$\forall\, i, j \in \{1, 2, \dots, n\} : \quad x_i + B[i, j] \quad \geq \quad x_j \tag{2}$$

   If there is no maximum latency between jobs i and j, then $B[i, j] = \infty$.

(a) Show how to model the latency constraints as a set of linear difference equations. That is, given $A[1 \,..\, n, 1 \,..\, n]$ and $B[1 \,..\, n, 1 \,..\, n]$, construct a matrix $C[1 \,..\, n, 1 \,..\, n]$ such that the following constraints are equivalent to Equations (1) and (2):

$$\forall\, i, j \in \{1, 2, \dots n\} : \quad x_i - x_j \quad \leq \quad C[i, j] \tag{3}$$

(b) Show that the Bellman-Ford algorithm, when run on the constraint graph corresponding to Equation (3), minimizes the quantity $(\max\{x_i\} - \min\{x_i\})$ subject to Equation (3) and the constraint $x_i \leq 0$ for all x_i.

(c) Give an efficient algorithm for minimizing the overall duration of the construction schedule. That is, given $A[1 \,..\, n, 1 \,..\, n]$ and $B[1 \,..\, n, 1 \,..\, n]$, choose $\{x_1, x_2, \ldots, x_n\}$ so as to minimize $\max\{x_i\}$ subject to the latency constraints and the constraint $x_i \geq 0$ for all x_i. Assume that an unlimited number of jobs can be performed in parallel.

(d) If the constraints are infeasible, we'd like to supply the user with information to help in diagnosing the problem. Extend your algorithm from (c) so that, if the constraints are infeasible, your algorithm prints out a set S of conflicting constraints that is minimal—that is, if any constraint is dropped from S, the remaining constraints in S would be feasible.

Problem 6-3. Honeymoon Hiking

Alice and Bob (after years of communicating in private) decide to get married and go hiking for their honeymoon. They obtain a map, which they naturally regard as an undirected graph $G = (V, E)$; vertices represent locations and edges represent trails. Some of the locations are bus stops, denoted by $S \subset V$.

Alice and Bob consider a hike to be ***romantic*** if it satisfies two criteria:

1. It begins and ends at different bus stops.
2. All of the uphill segments come at the beginning (and all of the downhill segments come at the end).[1]

Let $w(e) \in \mathbb{R}^+$ denote the length of a trail $e \in E$, and let $h(v) \in \mathbb{R}$ denote the elevation (height) of a location $v \in V$. You may assume that no two locations have exactly the same elevation.

(a) Give an efficient algorithm to find the *shortest* romantic hike for Alice and Bob (if any romantic hike exists).

(b) Give an efficient algorithm to find the *longest* romantic hike for Alice and Bob (if any romantic hike exists).

[1]After all, what is more frustrating than going uphill after you have started going downhill?