CSCI 665 Foundations of Algorithms

SPRING 2019

Lab Assignment III

Due on April 26, 2019 4:00 PM

Submission:
For each problem, you are required to write the Algorithm and Implement it.
Show working code for just one example – a page of text for Project 1;
Sample data given for Problem 2.
Use Dropbox to submit by 4:00 PM, 04/26/19.

Problem 1

Consider the problem of neatly printing a paragraph on a printer. The input text is a sequence of n words of length $l_1, l_2, \ldots, l_n$, measured in input characters. We want to print this paragraph neatly on a number of lines that hold a maximum of M characters each. Our criterion of "neatness" is as follows. If a given line contains words i through j and leave exactly one space between words, the number of extra space characters at the end of the line is,

$$\boldsymbol{M - j + i - \sum_{k=i}^{j} l_k}$$

We wish to minimize the sum, over all the lines except the last of the extra space characters at the ends of lines. Give a dynamic programming algorithm to print a paragraph of n words neatly on a printer. Analyse the running time and space requirements of your algorithm.
You may consider a page of text as input.

Please note the same problem is in the textbook by Algorithm Design by Kleinberg and Tardos
Chapter 6, Exercise Problem 6.

Problem 2

From textbook by Kleinberg and Tardos
Chapter 7, Exercise Problem 8
A PDF of the scanned exercise problem is in the following pages.

There is a *range parameter* r—a client can only be connected to a base station that is within distance r. There is also a *load parameter* L—no more than L clients can be connected to any single base station.

Your goal is to design a polynomial-time algorithm for the following problem. Given the positions of a set of clients and a set of base stations, as well as the range and load parameters, decide whether every client can be connected simultaneously to a base station, subject to the range and load conditions in the previous paragraph.

8. Statistically, the arrival of spring typically results in increased accidents and increased need for emergency medical treatment, which often requires blood transfusions. Consider the problem faced by a hospital that is trying to evaluate whether its blood supply is sufficient.

The basic rule for blood donation is the following. A person's own blood supply has certain *antigens* present (we can think of antigens as a kind of molecular signature); and a person cannot receive blood with a particular antigen if their own blood does not have this antigen present. Concretely, this principle underpins the division of blood into four *types*: A, B, AB, and O. Blood of type A has the A antigen, blood of type B has the B antigen, blood of type AB has both, and blood of type O has neither. Thus, patients with type A can receive only blood types A or O in a transfusion, patients with type B can receive only B or O, patients with type O can receive only O, and patients with type AB can receive any of the four types.[4]

(a) Let s_O, s_A, s_B, and s_{AB} denote the supply in whole units of the different blood types on hand. Assume that the hospital knows the projected demand for each blood type d_O, d_A, d_B, and d_{AB} for the coming week. Give a polynomial-time algorithm to evaluate if the blood on hand would suffice for the projected need.

(b) Consider the following example. Over the next week, they expect to need at most 100 units of blood. The typical distribution of blood types in U.S. patients is roughly 45 percent type O, 42 percent type A, 10 percent type B, and 3 percent type AB. The hospital wants to know if the blood supply it has on hand would be enough if 100 patients arrive with the expected type distribution. There is a total of 105 units of blood on hand. The table below gives these demands, and the supply on hand.

[4] The Austrian scientist Karl Landsteiner received the Nobel Prize in 1930 for his discovery of the blood types A, B, O, and AB.

blood type	supply	demand
O	50	45
A	36	42
B	11	8
AB	8	3

Is the 105 units of blood on hand enough to satisfy the 100 units of demand? Find an allocation that satisfies the maximum possible number of patients. Use an argument based on a minimum-capacity cut to show why not all patients can receive blood. Also, provide an explanation for this fact that would be understandable to the clinic administrators, who have not taken a course on algorithms. (So, for example, this explanation should not involve the words *flow*, *cut*, or *graph* in the sense we use them in this book.)

9. Network flow issues come up in dealing with natural disasters and other crises, since major unexpected events often require the movement and evacuation of large numbers of people in a short amount of time.

   Consider the following scenario. Due to large-scale flooding in a region, paramedics have identified a set of n injured people distributed across the region who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals, depending on where they are right now).

   At the same time, one doesn't want to overload any one of the hospitals by sending too many patients its way. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the injured people in such a way that the load on the hospitals is *balanced*: Each hospital receives at most $\lceil n/k \rceil$ people.

   Give a polynomial-time algorithm that takes the given information about the people's locations and determines whether this is possible.

10. Suppose you are given a directed graph $G = (V, E)$, with a positive integer capacity c_e on each edge e, a source $s \in V$, and a sink $t \in V$. You are also given a maximum s-t flow in G, defined by a flow value f_e on each edge e. The flow f is *acyclic:* There is no cycle in G on which all edges carry positive flow. The flow f is also integer-valued.