Engineering and Applied Science Programs for Professionals
Whiting School of Engineering
Johns Hopkins University
605.421 Foundations of Algorithms
Homework 6
Due at the end of Module 12

Total Points 100/100

Collaboration groups will be set up in Blackboard by the end of the week. Make sure your group starts an individual thread for each collaborative problem and subproblem. You are required to participate in each of the collaborative problem and subproblem.

Self-Study Problems

All of the following problems come from the textbook and have solutions posted on the web at http://mitpress.mit.edu/algorithms.

You are permitted to use this site to examine solutions for these problems as a means of self-checking your solutions. These problems will not be graded.

Problems: 15.2-5, 15.3-1, 15.4-4, 15-4, 16.1-4, 16.2-2, 16.2-7

Problems for Grading

1. **Problem 1 Chapter 15**
   20 Points Total

   Suppose you are consulting for a company that manufactures PC equipment and ships it to distributors all over the country. For each of the next n weeks, they have a projected supply s_i of equipment (measured in pounds) that has to be shipped by an air freight carrier. Each weeks supply can be carried by one of two air freight companies, A or B.

   - Company A charges a fixed rate r per pound (so it costs $r \times s_i$ to ship a week's supply s_i).

   - Company B makes contracts for a fixed amount c per week, independent of the weight. However, contracts with company B must be made in blocks of four consecutive weeks at a time.

   A schedule for the PC company is a choice of air freight company (A or B) for each of the n weeks with the restriction that company B, whenever it is chosen, must be chosen for blocks of four contiguous weeks in time. The cost of the schedule is the total amount paid to company A and B, according to the description above.

   Give a polynomial-time algorithm that takes a sequence of supply values $s_1, s_2, ..., s_n$ and returns a schedule of minimum cost. For example, suppose $r = 1, c = 10$, and the sequence of values is

   $$11, 9, 9, 12, 12, 12, 12, 9, 9, 11.$$

   Then the optimal schedule would be to choose company A for the first three weeks, company B for the next block of four contiguous weeks, and then company A for the final three weeks.

2. **Problem 2 Chapter 29**
   20 Points Total

   CLRS 29.4-3 Write down the dual of the maximum-flow linear program, as given in Equations (29.47) - (29.50) on page 860 of the textbook. Explain how to interpret this formulation as a minimum-cut problem.

3. **Problem 3** ***Note this is a Collaborative Problem***
   20 Points Total

   As some of you know well, and others of you may be interested to learn, a number of languages (including Chinese and Japanese) are written without spaces between the words. Consequently, software that works with text written in these languages must address the word segmentation problem - inferring likely boundaries between consecutive words in the text. If English were written without spaces, the analogous problem would consists of taking a string like "meetateight" and decide that the best segmentation is "meet at eight" (and not "me et at eight" or "meet ate ight" or any huge number of even less plausible alternatives). How could we automate this process?

   A simple approach that is at least reasonably effective is to find a segmentation that simply maximizes the cumulative "quality" of its individual constituent works. Thus, suppose you are given a black box that, for any string of letters $x = x_1x_2...x_k$, will return a number $quality(x)$. This number can be either positive of negative; large numbers correspond to more plausible English words. (so $quality$("me") would be positive while $quality$("ight") would be negative.)

   Given a long string of letters $y = y_1y_2...y_n$, a segmentation of y is a partition of its letters into contiguous blocks of letters, each block corresponds to a word in the segmentation. The total quality of a segmentation is determined by adding up the qualities of each of its blocks. (So we would get the right answer above provided the $quality$("meet") + $quality$("at") + $quality$("eight") was greater than the total quality of any other segmentation of the string.) Give an efficient algorithm that takes a string y and computes a segmentation of maximum total quality. You can treat a single call to the black box computing $quality$("meet"x) as a single computational step. Prove the correctness of your algorithm and analyze its time complexity.

4. **Problem 4 Chapter 29** ***Note this is a Collaborative Problem***
40 Points Total 10 Each

In the course content, it is explained how to solve two-player zero-sum games using linear programming. One of the games described is "Rock-Paper-Scissors." In this problem the game is examined closer. Assume the the following is a "loss" matrix for Player 1, e.g., the matrix shows how much Player 1 has lost rather than gained so the sign is reversed:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

(a) What is the expected loss for Player 1 when Player 1 plays a mixed strategy $\mathbf{x} = (x_1, x_2, x_3)$ and Player 2 plays a mixed strategy $\mathbf{y} = (y_1, y_2, y_3)$?

(b) Show that Player 1 can achieve a *negative* expected loss (i.e., an expected gain) if Player 2 plays any strategy other than $\mathbf{y} = (y_1, y_2, y_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

(c) Show that $\mathbf{x} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $\mathbf{y} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ from Nash equilibrium.

(d) Let $\mathbf{x} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ as in part (c). Is it possible for $(\mathbf{x}, \mathbf{y})$ to be a Nash equilibrium for some mixed strategy $\mathbf{y}' \neq (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$? Explain.