CPSC-500 Fundamentals of Algorithm Design and Analysis Winter 2013

Homework # 4

Due in class on Wednesday, Nov 13

You are free to discuss problems with your fellow students. However, your actual homework submission must be prepared on your own. At the top of the first page of every homework submission you must provide a statement about external resources used in the preparation of your submission. This must clearly acknowledge all resources (including books, websites and discussions with fellow students) that you have used. Submissions missing this statement will not be graded.

1. Two halloween trick-or-treaters share a haul with n treats. They have agreed on a value for each treat, given by the positive integers $x_1, x_2, \ldots, x_n$. They want to split the treats into two sets of equal value.

   Design an algorithm that, given the values $x_1, x_2, \ldots, x_n$, determines whether it is possible for the trick-or-treaters to fairly split the treats and if so, outputs a fair split. Your algorithm should run in time $O(nT)$, where T is the sum of the x_i's.

2. To do this problem, you'll need to read part of the following paper: "A dynamic programming algorithm to find all solutions in a neighbourhood of the optimum" by Waterman, Michael S. and Byers, Thomas H., *Mathematical Biosciences*, 77:179-188, 1985. It is easy to find online.

   The *Algorithm* section of the paper first describes a dynamic programming solution to the problem of finding the shortest path from every node of a directed acyclic graph to a single destination N.

   Suppose that $f(1)$ is the length of the shortest path from node 1 to node N of the graph. The section goes on to describe, informally, how to find all paths of length at most $f(1) + e$, for a given fixed positive constant e, by "performing a depth-first search with stacking".

   Your task is to write pseudocode that makes precise the informal description provided in the paper for finding all paths from node 1 to node N that have length at most $f(1) + e$.

3. A substring α of a string S is called a *tandem array* of β—called the base—if α consists of more than one consecutive copy of β. For example, if $S = xyzabcabcabcabcpq$ then $\alpha = abcabcabcabc$ is a tandem array of base $\beta = abc$. (Note that S also contains a tandem array of $abcabc$, i.e. a tandem array with a longer base.) A *maximal* tandem array is a tandem array that cannot be extended either left or right. Given the base β, a tandem array of β in S can be described by two numbers (s, k) giving its starting location in S and the number of times β is repeated. A tandem array is an example of a repeated substring; identification of tandem arrays arises in analysis of genomic DNA molecules.

   (a) Give an example to show that two maximal tandem arrays of a given base β can overlap.

   (b) Give an $O(n)$ time algorithm that takes a string S of length n and a base β of length at most n as input, finds every maximal tandem array of β in S, and outputs the pair (s, k) for each occurrence.

4. The following problem is motivated by the goal of aligning bacterial DNA sequences, which are often organized into circular molecules. Given two strings x and y of lengths n and m respectively, there are n circular shifts of x and m circular shifts of y, and so there are mn pairs of circular shifts. For example, the three circular shifts of *abc* are *abc*, *bca*, and *cab*.

   Describe an efficient algorithm to compute the best global alignment of all pairs of circular shifts of input strings x and y over the alphabet $\{A, C, G, T\}$. Assume that the alignment is scored using a matrix δ just as in Exercise 6.26 (page 197) of the Algorithms textbook of Dasgupta, Papadimitriou and Vazirani, or as in the class lecture notes from October 21.