Design and Analysis of Algorithms
Assignment 3

October 8, 2012

Marks: 30 — Due date: 19th October

1. You have seen how to find edge-disjoint paths from s to t in a graph using network flows. Using similar method, show how to find vertex-disjoint paths from s to t. 4 marks

   (Hint: Unlike in the case of edge-disjoint paths, you need to modify the graph here.)

2. Construct the dual of the following linear programs: 4 marks

   Maximize $3x_1 + 4x_2 + 2x_3$ subject to

$$\begin{aligned} 5x_1 + 2x_2 &\le 10 \\ 3x_2 + 4x_3 &\le 7 \\ x_1 + x_2 + x_3 &= 3 \end{aligned}$$

   Minimize $2x_1 + 5x_2 + 6x_3$ subject to

$$\begin{aligned} 3x_1 + 4x_2 + 3x_3 &\ge 8 \\ x_1 + x_2 + x_3 &\ge 3 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

3. A *vertex cover* is a set of vertices $S \subseteq V$ in a graph such that each edge has at least one of its end-points in S. The vertex cover problem is to find an S of minimum cardinality.

   Write an LP for this problem and construct its dual. Using strong duality, show that in a bipartite graph, the size of a minimum vertex cover is equal to the size of a maximum matching. 6 marks

4. We are given a string $S = a_0a_1 \ldots a_{n-1}$ of zeros and ones of length n. We have to find if there exist three ones in S, say a_i, a_j, a_k, such that $k - j = j - i \ge 1$. Such a triple of ones will be referred to as a *well-spaced triple*. The string 10100101 does not have a well-spaced triple whereas the string 010100110011 does. 8 marks

   (a) Give a simple $O(n^2)$ algorithm to find a well-spaced triple in a given string S.

   The goal in the next parts is to use FFT to find an $O(n \log n)$ time algorithm.

   (b) Define *convolution* of two strings $a_0a_1 \ldots a_{n-1}$ and $b_0b_1 \ldots b_{m-1}$ to be another string $c_0c_1 \ldots c_{n+m-2}$ where $c_k = \sum_{i,j:i+j=k} a_ib_j$. In the given problem, suppose we convolve S with itself, $P = CONVOLVE(S, S)$. Give an interpretation of P_i in the cases when i is odd and even.

   (c) Use your interpretation to count the number of well-spaced triples in $O(nlogn)$ time.

   (d) Give an algorithm to find a well-spaced triple in $O(n)$ time, given the convolution P.

   Open question: Give an algorithm to find *all* well-spaced triples in $O(n + T)$ time given P, where T is the number of well-spaced triples in S. Note: By above algorithm, we need $O(nT)$ time.

5. Problem 2.30 page 88-89 from Dasgupta et al. 8 marks