CSL 356 Algorithm Design and Analysis

Major, Sem I 2006-07, Max 90, Time 2 hrs

Note Every algorithm must be accompanied by proof of correctness, time and space complexity. You can however quote any result covered in the lectures without proof. Problems 1-7 carry 10 marks each and Problem 8 is (5+15).

1. You are given n pairs of nuts and bolts of distinct sizes. You can find out a matching pair by actually testing if the bolt and nut physically fit (you can also determine if the nut or the bolt is oversized). Each such test is a primitive operation. Design a strategy to find all the matching pairs using a minimal number (asymptotically) of primitive operations. Your algorithm should use $o(n^2)$ operations.

2. Given a set S of n numbers (not necessarily integers), design an efficient algorithm to determine if there are $a, b, c \in S$ such that $a + b = c$? (No credit for $\theta(n^3)$ algorithm).

3. Given two strings each of length n describe an efficient algorithm to determine if they are cyclic shifts of eachother. For example the string **aceabe** can be obtained from **eabeac** by a cyclic shift.

4. Given a weighted graph $G = (V, E)$ and an edge $e \in E$, describe an $O(m+n)$ algorithm to determine if e is an edge of the MST of G. (You may assume that there is a unique MST).

5. Consider an $n \times n$ array of real numbers rolled into a cylinder (see Figure 1). A path begins from the left end and goes to the right end with the restriction that from a given square it is possible to go to one of the three neighbouring squares in the next column. A path may begin at any square on the left-most column and can end at any square on the rightmost column. The cost of a path is the sum of values on the squares that the path traverses.
   Describe an $O(n^2)$ algorithm to find the shortest path.

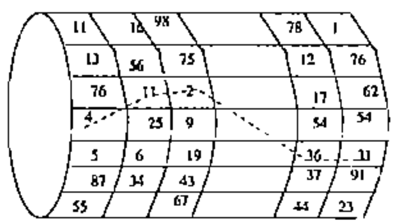

Figure 1: The cost of the dotted path is 4+11+2+ ...+36+31

6. Consider the following greedy approach to finding a minimum vertex cover of a given graph (a vertex cover must include at least one end point of every edge). Initialise $T = \Phi$, pick an (arbitrary) edge (u, v) and include u, v in T. Delete all edges that are incident on u and v (they are covered by u, v) and repeat the above procedure on the remaining graph till all edges are deleted. Output the set T as a vertex cover. If O is a minimum vertex cover, can you bound $\frac{|T|}{|O|}$?

7. Given an unweighted bipartite graph, we want to find the maximum matching using Ford-Fulkerson method for finding maxflows. Describe the approach in details and analyse the running time.

8. (i) Given two decision problems Π_1 and Π_2, both of which have polynomial time algorithms, show that $\Pi_1 \leq_{poly} \Pi_2$. Assume that you know at least one instance I_Y of Π_2 that has answer **YES** and at least one instance I_N of Π_2 that has answer **NO**.
   (ii) Show that the following problem is NP Complete.
   Given a graph $G = (V, E)$ and an integer $k \leq |V|$, we want to find a subset $V' \subset V$, $|V'| \leq k$ such that every vertex $v \in V - V'$ has a neighbour in V'.
   Hint: Use Vertex cover and the following construction - replace every edge $e_i = (e_{i1}, e_{i2})$ by a triangle e_i, e_{i1}, e_{i2}.