COL 351

TUTORIAL SHEET 2

1. You are given a set of intervals on a line segment. You wish to color these segments such that no two overlapping segments get the same color. Devise a greedy algorithm for coloring the intervals which uses as few colors as possible.

2. Consider the problem of making change for n rupees using the fewest number of coins. Suppose that the available coins are in denominations that are powers of c, i.e., the denominations are $c^0, c^1, \ldots, c^k$ for some integer $c > 1$ and $k \geq 1$. Show that the following greedy algorithm always yields an optimal solution – pick as many coins of denomination c^k as possible, then pick as many coins of denomination c^{k-1} and so on.

3. **(KT-Chapter 4)** Suppose you are given an undirected graph G, with edge weights that you may assume are all distinct. G has n vertices and m edges. A particular edge e of G is specified. Give an algorithm with running time $O(m + n)$ to decide whether e is contained in a minimum-weight spanning tree of G.

4. **(KT-Chapter 4)** Suppose you have n video streams that need to be sent, one after another, over a communication link. Stream i consists of a total of b_i bits that need to be sent, at a constant rate, over a period of t_i seconds. You cannot send two streams at the same time, so you need to determine a schedule for the streams: an order in which to send them. Whichever order you choose, there cannot be any delays between the end of one stream and the start of the next. Suppose your schedule starts at time 0 (and therefore ends at time $\sum_{i=1}^{n} t_i$ whichever order you choose). We assume that all the values b_i and t_i are positive integers. Now, because you're just one user, the link does not want you taking up too much bandwidth – so it imposes the following constraint, using a fixed parameter r:

   (*) For each natural number $t > 0$, the total number of bits you send over the time interval from 0 to t cannot exceed rt.

   Note that this constraint is only imposed for time intervals that start at 0, not for time intervals that start at any other value. We say that a schedule is valid if it satisfies the constraint (*) imposed by the link. The problem is: Given a set of n streams, each specified by its number of bits b_i and its time duration t_i, as well as the link parameter r, determine whether there exists a valid schedule.

   Example. Suppose we have $n = 3$ streams, with $(b_1, t_1) = (2000, 1), (b_2, t_2) = (6000, 2), (b_3, t_3) = (2000, 1)$, and suppose the link's parameter is $r = 5000$. Then the schedule that runs the streams in the order 1, 2, 3, is valid, since the constraint (*) is satisfied:
   $t = 1$: the whole first stream has been sent, and $2000 < 5000 \cdot 1$

$t = 2$: half the second stream has also been sent, and $2000 + 3000 < 5000 \cdot 2$. Similar calculations hold for $t = 3$ and $t = 4$.

(a) Consider the following claim:
Claim: There exists a valid schedule if and only if each stream i satisfies $b_i \leq rt_i$.
Decide whether you think the claim is true or false, and give a proof of either the claim or its negation.

(b) Give an algorithm that takes a set of n streams, each specified by its number of bits b_i and its time duration t_i, as well as the link parameter r, and determines whether there exists a valid schedule. The running time of your algorithm should be polynomial in n. You should prove that your algorithm works correctly, and include a brief analysis of the running time.

5. **(KT-Chapter 4)** Timing circuits are a crucial component of VLSI chips; here's a simple model of such a timing circuit. Consider a complete binary tree with n leaves, where n is a power of two. Each edge e of the tree has an associated length l_e, which is a positive number. The distance from the root to a given leaf is the sum of the lengths of all the edges on the path from the root to the leaf. The root generates a clock signal which is propagated along the edges to the leaves. We?ll assume that the time it takes for the signal to reach a given leaf is proportional to the distance from the root to the leaf. Now, if all leaves do not have the same distance from the root, then the signal will not reach the leaves at the same time, and this is a big problem: we want the leaves to be completely synchronized, and all receive the signal at the same time. To make this happen, we will have to increase the lengths of certain edges, so that all root-to-leaf paths have the same length (we're not able to shrink edge lengths). If we achieve this, then the tree (with its new edge lengths) will be said to have zero skew. Our goal is to achieve zero skew in a way that keeps the sum of all the edge lengths as small as possible. Give an algorithm that increases the lengths of certain edges so that the resulting tree has zero skew, and the total edge length is as small as possible.

6. **(KT-Chapter 4)** Given a list of n natural numbers $d_1, d_2, \ldots, d_n$, show how to decide in polynomial time whether there exists an undirected graph $G = (V, E)$ whose node degrees are precisely the numbers $d_1, \ldots, d_n$. (That is, if $V = \{v_1, \ldots, v_n\}$, then the degree of v_i should be exactly d_i.) G should not contain multiple edges between the same pair of nodes, or "loop" edges with both endpoints equal to the same node.