COL 351

TUTORIAL SHEET 1

1. (KT-Chapter 4) Let us consider a long, quiet country road with houses scattered very sparsely along it. (We can picture the road as a long line segment, with an eastern endpoint and a western endpoint.) Further, let's suppose the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within 4 kilometers of one of the base stations. Give an efficient algorithm that achieves this goal, using as few base stations as possible. Prove the correctness of your algorithm.

   Solution: There is a simple greedy algorithm here. Let h denote the left-most house. Then we place a base station 4km to the right of h. Now remove all houses which are covered by this base station, and repeat. It is easy to show (by induction) that if the algorithm locates base stations at $b_1, \ldots, b_k$, and some other algorithm (which could be the optimal algorithm) places base stations at $b'_1, \ldots, b'_{k'}$ (from left to right), then $b_1 \geq b'_1, b_2 \geq b'_2$, and so on. Therefore $k \leq k'$.

2. (KT-Chapter 4) Consider the following variation on the Interval Scheduling Problem from lecture. You have a processor that can operate 24 hours a day, every day. People submit requests to run daily jobs on the processor. Each such job comes with a start time and an end time; if the job is accepted to run on the processor, it must run continuously, every day, for the period between its start and end times. (Note that certain jobs can begin before midnight and end after midnight; this makes for a type of situation different from what we saw in the Interval Scheduling Problem.)

   Given a list of n such jobs, your goal is to accept as many jobs as possible (regardless of their length), subject to the constraint that the processor can run at most one job at any given point in time. Provide an algorithm to do this with a running time that is polynomial in n, the number of jobs. You may assume for simplicity that no two jobs have the same start or end times.

   Example: Consider the following four jobs, specified by (start-time, end-time) pairs: (6 pm, 6 am), (9 pm, 4 am), (3 am, 2 pm), (1 pm, 7 pm). The unique solution would be to pick the two jobs (9 pm, 4 am) and (1 pm, 7 pm), which can be scheduled without overlapping.
   Solution: This is like the interval scheduling problem discussed in class except that jobs are periodic. One way of thinking about this is that time is not like an interval, but is a circle (think of it as the circle in a clock). Now, each job corresponds to an interval in this circle. Now, suppose we *knew* the job which are being done at midnight – call this j. Then, we could remove all jobs overlapping with j, and solve the remaining problem. The remaining problem looks like the interval scheduling problem done in

class (and so can be solved using a greedy algorithm). However, we do not know the job j. One way out is to try out all possibilities for such a job: for each job which contains the mid-night time, we select it and solve the resulting interval scheduling problem. Finally, we pick the best such solution.

3. You are given a line with n points, labeled 1 to n, marked on it. You are also given a set of intervals $I_1, \ldots, I_k$, where interval I_i is of the form $[s_i, e_i]$, $1 \leq s_i \leq e_i \leq n$. Find a set of points X of smallest cardinality such that each interval contains at least one point from X.

   Solution: Sort the intervals in increasing order of e_i. Select the first point as the right end-point of the first interval in this order. Remove all intervals which intersect with this point, and repeat. Proof of correctness is similar to the interval scheduling problem discussed in class. If the algorithm picks points $p_1 < p_2 < \cdots < p_k$, and optimum solution picks points $q_1 < q_2 < \cdots, q_s$, then prove by induction that $p_i \geq q_i$, and so $k \leq s$.

4. You are given two sets X and Y of n positive integers each. You are asked to arrange the elements in each of the sets X and Y in some order. Let x_i be the i^{th} element of X in this order, and define y_i similarly. Your goal is to arrange them such that $\Pi_{i=1}^{n} x_i^{y_i} = x_1^{y_1} \times x_2^{y_2} \times \cdots \times x_n^{y_n}$ is maximized. Give an efficient algorithm to solve this problem. Prove correctness of your algorithm.

   Solution: Arrange the elements of X in decreasing order of x_i values – let this ordering be $1, 2, \ldots, n$. Do the same for Y, and let the ordering be $1, 2, \ldots, n$. These are the orderings of X and Y produced by the greedy algorithm. You can always assume that the optimum solution orders X as $1, \ldots, n$ as well. Now if it does not order Y as $1, \ldots, n$, then there must be two consecutive elements in the ordering of Y, say i_1, i_2, such that $i_1 > i_2$. Now show that by reversing the ordering, you get a better solution. And then argue as done in class: we reduce the number of inversions with respect to our solution.

5. Suppose you want to go from city A to city B on a long highway. Once you fill your car tank to full capacity, it can travel D kilometres. There are several locations on the highway which have petrol pumps. Assume that there is a petrol pump at the start of the highway, and every stretch of length D on the highway has at least one location with a petrol pump. Given the location of these petrol pumps, devise a strategy for traveling from A to B so that you will have to make as few stops for filling petrol as possible.

   Solution: Solution is again a greedy algorithm. Let p be the last petrol station at which we filled petrol. Look at the segment of length D with left end-point at p, and choose the next petrol station as the last one in this segment, and so on. Again, the proof is like Question 1 above: if the algorithm fills petrol at $p_1, \ldots, p_k$ (from left to right), and optimum does this at $q_1, \ldots, q_s$, then $p_i \geq q_i$.

   Another way of thinking about this problem is to *reduce* it to Problem 1 above: for each petrol station, draw an interval of length D with this petrol station as its right end-point.