Algorithms - Exercise 2

Due Sunday 7/11 24:00

1. Give an efficient algorithm for the following problem: We are given n intervals on a circle. We want to select a maximal number of disjoint intervals. (Notice that this is the activity selection problem, if we glue the past with the future.)

2. In the targil we considered the following problem: the input is n lectures $L_1, \ldots, L_n$ with their start and finish times $(s_1, f_1), \ldots, (s_n, f_n)$. There is an unlimited number of lecture halls, and the objective is to schedule the lectures in the halls such that no two lectures in the same hall overlap, and to do that using as few halls as possible. We showed a greedy choice that finds an optimal solution. Let us try different choices, or in other words, do the following algorithms work? (prove your answers or give a counter example)

   (a) Order the lectures by their *finish* times in an increasing order. At stage i schedule the lecture with the i'th least finish time to the first hall that can accommodate it.

   (b) Schedule in the first hall as many lectures as possible (using the algorithm that was shown in class), continue with the remaining lectures on the second hall and so on until there are no more lectures to schedule.

   A *matroid* is an ordered pair $M = (S, I)$ satisfying the following conditions:

   (a) S is a nonempty set.

   (b) I is a nonempty hereditary family of subsets of S.

   (c) If $A \in I$, $B \in I$, and $|A| < |B|$, then there is some element $x \in B - A$ such that $A \cup \{x\} \in I$. (This condition is called the *exchange property*.)

3. We've seen in class that the two following families of sets are hereditary. Are they matroids?

   (a) Solutions to the activity allocation problem

   (b) Solutions to the 0-1 knapsack problem

4. Let $a_1, \ldots, a_n$ and N be natural numbers. Consider the problem of minimizing $\sum_{i=1}^{n} b_i$ while $\sum_{i=1}^{n} a_i b_i = N$. ($\{b_i\}$ non-negative integers) Assume that $a_1 = 1$

   (a) Write down the greedy (not necessarily working) algorithm for the problem.

   (b) Show that for $a_1 = 1$, $a_2 = 5$, $a_3 = 10$ and any N the greedy algorithm gives the right answer.

   (c) Show that for any natural numbers c, n if $a_i = c^{i-1}$ the greedy algorithm gives the right answer for every N.

   (d) (Reshut) Show that for every $N > 10$ there is a way to define the $\{a_i\}$ to fool the greedy algorithm.