MAT-72006 Advanced Algorithms and Data Structures
October 27, 2016
HW 7: 16 Greedy Algorithms

1.

Determine an LCS of $\langle 1,0,0,1,0,1,0,1\rangle$ and $\langle 0,1,0,1,1,0,1,1,0\rangle$.

2.

Consider a computational problem and a greedy approach for solving it. Regardless of whether the approach is correct (i.e., always gives an optimal solution), why could it be worth considering such an approach? For example, you might consider the question from a practical viewpoint.

3.

Recall the activity-selection problem in which the goal is to find a maximum-size set of mutually compatible activities. Further, consider the following greedy approaches:

- always selecting the activity of least duration among those that are compatible with previously selected activities,
- always selecting the compatible activity that overlaps the fewest other remaining activities, and
- always selecting the compatible remaining activity with the earliest start time.

Give counterexamples to show that neither of the three mentioned greedy approaches always finds an optimal solution.

4.

In the *heaviest subset problem*, we are given a set U of integers and an integer k. The task is to output a set $S \subseteq U$ of size k such that the sum of elements in S is as large as possible. Give a greedy algorithm for the problem. Prove that your algorithm is correct and runs in polynomial time.

5.

In the *red-blue division problem*, we are given a set $U = \{1,\ldots,n\}$ and a collection of subsets $\mathcal{F} \subseteq 2^U$ each of size at least two. The goal is to color each integer in U either red or blue such that the number of sets in $\mathcal{F}$ containing both colors is maximized (we call such a set *two-colored*). For example, if $U = \{1,2,3,4\}$ and $\mathcal{F} = \{\{1,3\},\{1,2,3\},\{3,4\},\{2,4\}\}$, then by coloring 1 with red, 2 with blue, 3 with blue, and 4 with red, we two-color each member of $\mathcal{F}$.

Consider the following greedy approach for the problem: process each element u_i of U (in ascending order), and give u_i the color that maximizes the number of new sets two-colored (if no new set can get two-colored, or equally many new sets would get two-colored, color u_i with red). Does this greedy approach always find an optimal solution? Prove your answer.