Below is a set of practice problems on designing and proving the correctness of greedy algorithms, to help you check your understanding of the approach we’ve discussed in class.

You can find solutions to these problems on the course web site. We’re also willing to listen to you describe your solutions or to look at your writeups during office hours.

Note: If you want to discuss these problems, I expect you to give correctness and complexity proofs for your algorithms here, just as on the actual homework. Efficiency matters!

Practice Problems

1. Given a set \( \{x_1 \leq x_2 \leq \ldots \leq x_n\} \) of points on the real line, give an algorithm to determine the smallest set of unit-length closed intervals that contains all of the points. A closed interval includes both its endpoints; for example, the interval \([1.25, 2.25]\) includes all \(x_i\) such that \(1.25 \leq x_i \leq 2.25\).

2. Suppose you are driving from St. Louis to Denver along I-70. Your gas tank, when full, holds enough gas to travel \(m\) miles, and you have a map that gives distances between gas stations along the route. Let \(d_1 < d_2 < \cdots < d_n\) be the locations of all the gas stations along the route, where \(d_i\) is the distance from St. Louis to the gas station.

Your goal is to make as few gas stops as possible along the way. Give the most efficient algorithm you can to determine at which gas stations you should stop, and prove that your strategy yields an optimal solution. Be sure to give the time complexity of your algorithm as a function of \(n\).

3. You are given a sequence of \(n\) songs, where the \(i\)th song is \(\ell_i\) minutes long. You want to place all of the songs onto a collection of CD’s, each of which can hold \(m\) minutes. Furthermore,

   (1) The songs must be recorded in the order given: song 1, song 2, \ldots, song \(n\).

   (2) All songs must be included.

   (3) No song may be split across CDs.

Give an algorithm to place songs on CDs so as to minimize the number of CDs needed.

4. Consider the following scheduling problem. You have \(n\) jobs; each job \(i\) has a release time \(s_i\) and a running time \(\ell_i\). Job \(i\) may be scheduled starting at any time \(\geq s_i\) and must run for total time \(\ell_i\) in order to complete.

To make life interesting, we will make this scheduling problem preemptive: any job may be suspended at any time and resumed later. For example, if we have two jobs with \(s_1 = 2, \ell_1 = 5, s_2 = 0, \) and \(\ell_2 = 3\), then one feasible schedule (of many) would run the second job from time 0 to 2, then the first job from time 2 to 7, and finally the second job from 7 to 8.

The goal is to find a schedule that minimizes \(\sum_{i=1}^{n} C_i\), where \(C_i\) is the time at which job \(i\) completes. In the above example, \(C_1 = 7\) and \(C_2 = 8\). Give an optimal algorithm for this problem.