CS 473: Algorithms, Fall 2010
HBS 5

Problem 1. [Largest Contiguous Sum]
Suppose you are given an array $A[1...n]$ of numbers, which may be positive, negative, or zero. Describe an algorithm that finds the largest of elements in a contiguous subarray $A[i...j]$. For example, if the array contains the numbers $(-6, 12, -7, 0, 14, -7, 5)$, then the largest sum of any contiguous subarray is $19 = 12 - 7 + 0 + 14$.

Problem 2. [Hotel Stops]
You are going on a long trip. You start on the road at mile post 0. Along the way, there are n hotels, at mile posts $a_1 < a_2 < \cdots < a_n$, where each a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance a_n), which is your destination.

You'd ideally like to travel 200 miles a day, but this many not be possible (depending on the spacing of the hotels). If you travel x miles during a day, the *penalty* for that day is $(200 - x)^2$. You want to plan your trip so as to minimize the total penalty–that is, the sum over all travel days, of the daily penalties.

Give an efficient algorithm that determines the optimal sequence of hotels at which to stop.

Problem 3. [Longest Common Subsequence]
Let $A[1...m]$ and $B[1...n]$ be two arbitrary arrays. A *common subsequence* of A and B is another sequence that is a subsequence of both A and B. Describe an efficient algorithm to compute the length of the *longest* common subsequence of A and B.

A subsequence is anything obtained from a sequence by extracting a subset of elements, but keeping them in the same order; the elemtns of the subsequence need not be contiguous in the original sequence. For example, the strings *C*, *DAMN*, and *YAIOAI*, and *DYNAMICPROGRAMMING* are all subsequences of the sequence *DYNAMICPROGRAMMING*.