HW 4 (due Tuesday, at noon, February 25, 2014)

CS 473: Fundamental Algorithms, Spring 2014

Make sure that you write the solutions for the problems on separate sheets of paper. Write your name and netid on each sheet.

Collaboration Policy: The homework can be worked in groups of up to 3 students each.

---

1. (35 PTS.) Planning a road trip.

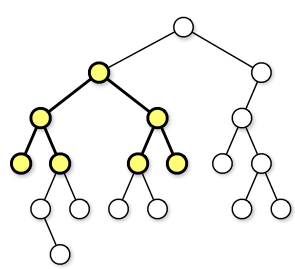
You are going on a long trip. You start on the road at mile post 0. Along the way there are n hotels, at mile posts $a_1 < a_2 < \ldots < a_n$, where a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance a_n) which is your destination. You would ideally like to travel 350 miles a day, but this may not be possible (depending on the hotel locations). If you travel $x \geq 0$ miles during a day, the penalty for that day is $(350 - x)^4$; note that x can more than 350 as well. 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. Describe an algorithm that determines the optimal sequence of hotels at which to stop so as to minimize the overall penalty.

2. (35 PTS.) Splitting a sequence.

Let $a_1, a_2, \ldots, a_n$ be a sequence of numbers (could be positive or negative) and let B be a number. You wish to decide whether the sequence can be split/partitioned into contiguous subsequences such that cube of the sum of numbers in each subsequence is at most B. For instance if the sequence is $2, 10, -1, 11$ and $B = 100$ then answer is no because the smallest value that can be obtained for the subsequence that contains 11 is $(11 - 1)^3 = 1000$. The answer is yes if $B = 1000$ where the sequence is split into three subsequences: [2], [10] and $[-1, 11]$. Describe an algorithm for this problem. Your algorithm, if it outputs yes, should also output a split of the sequence that shows the correctness of the answer.

3. (30 PTS.) Complete subtree.

For this problem, a *subtree* of a binary tree means any connected subgraph. A binary tree is *complete* if every internal node has two children, and every leaf has exactly the same depth. Describe and analyze a recursive algorithm to compute the largest complete subtree of a given binary tree. Your algorithm should return the root and the depth of this subtree.



The largest complete subtree of this binary tree has depth 2.