

CS 473: Algorithms, Fall 2009

HW 6 (due Tuesday, October 20th in class)

This homework contains three problems. **Read the instruction for submitting homework on the course web page.** In particular, *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: For this home work, students can work in groups of up to 3 members each. Each group submits only one written solution (some groups will do an oral presentation. Indicate your group members on the homework (netids are needed)).

1. (25 pts) Recall from class the idea of Path-Compression in the Union-Find data structure for maintaining disjoint sets. Prove that if Path-Compression is used then the total time for k operations in which all the `union` operations precede all the `find` operations, is $O(k)$. (The cost of initializing the data structure via `makeUnionFind` is not part of the cost of the k operations). Note that the running time is independent of the number of elements n .
2. (30 pts) Problem 6.11 from the text book.
3. (25 pts) 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 $a_1 < a_2 < \dots < a_n$, 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 200 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 $(200 - x)^2$; note that x can more than 200 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 a dynamic programming based algorithm that determines the optimal sequence of hotels at which to stop so as to minimize the overall penalty.

(20 pts) Implement your algorithm in C or C++. The input to your program will be a sequence of integers $n a_1 a_2 \dots a_n$ on a single line that are separated by white space (don't worry about error checking the input). The output should be the optimal penalty for the trip and the indices of the hotels to stop at. We want you to use a recursive function `ComputePenalty` with memoization and which runs in polynomial time. Your program should consist of only two functions: a main program that reads the input, sets up a couple of arrays, calls `ComputePenalty` and returns the output, and a recursive function `ComputePenalty`. You need to submit a print out of the code along with the output of your code on a set of inputs that we will provide on the website. You should not submit the code itself.