SOLUTIONS FOR SACO ONLINE CAMP, 2008

MAX RABKIN

First off, I'd like to thank everybody for participating. I'd also like to thank Bruce, Carl and Marco for helping me run this camp, and for writing and choosing the problems.

Your first task in a contest is to read the problems and rank them by difficulty. The problems in the SACO and IOI are *not* in order. Even if they were, your different strengths mean you might put them in a different order (I found "walls" as the easiest, and "area" the hardest, but "change" had the highest average score and "walls" the lowest).

After ordering the problems, solve the *easiest one first*. Don't think you'll do the hard one and then do the other two in the last half hour: you might never solve the hard one, and then not have time to debug the two you could have solved.

The next step, now, to read and understand these solutions. Implement them. Test them. If you can't get 100%, ask for help.

1. Making Change

We can use dynamic programming to find the smallest number of coins needed to make a certain amount. We populate a two-dimensional array using the relation

$$c_{a,n} = \begin{cases} 0 & \text{if } a = 0 \\ \infty & \text{if } i = 0, a \neq 0 \\ \min\left\{c_{a,n-1}, c_{a-v_n,n}\right\} & \text{otherwise} \end{cases}$$

where $c_{a,n}$ is the number of coins needed to make a cents using the first n coin types, and v_n is the value of the nth coin type. So $c_{a,N}$ is the smallest number of coins needed for a cents. The last case basically says that to make up a value, we either use the current coin together with the coins for a smaller amount, or we don't use the current coin.

Next, we prove that if a coin system (with largest coin m) is non-greedy, there is a value less than $2m$ where the greedy method and the optimal method give a different answer. The proof is by contradiction.

Let x be the smallest value where the greedy and optimal method differ.

The optimal solution for x includes some coin c along with the optimal solution for $x - c$. However, since x is minimal, the optimal solution for $x - c$ is greedy.

Assume $x \geq 2m$ (!). Then, since $c \leq m$, $x - c \geq m$, so the greedy (and thus also the optimal) solution for $x - c$ includes m. So the optimal solution for x also includes m. Since the greedy and optimal solutions both contain m, but they differ, the solutions for $x - m$ must differ. This means that x is not minimal, which is a contradiction.

So our assumption is false, and $x < 2m$.

So the solution to this problem is simply to test whether the greedy and optimal solutions differ for any amount less than $2m$

2. Blow-torching walls

This question can be modelled as a weighted graph, where it costs 1 unit to move through a non-wall and $c + 1$ to walk through a wall. We let c be more than the length of the longest path $WH + 1$ suffices. We use Dijkstra's algorithm (see any algorithms textbook or Wikipedia for a description) to find the cost p of the shortest path.

The number of blow-torched walls is $\lfloor \frac{p}{c} \rfloor$; the number of steps taken is $p \bmod c$.

3. Area

This problem can be solved in $\mathcal{O}\left(NH\right))$ time, where H is the number of rows. We construct a sorted list of rectangles, with each rectangle appearing twice: once sorted by its left coordinate and once by its right.

For each row, we initialise a counter to zero and scan through this list, finding the rectangles which cover this row. Every time we encounter a left edge covering this row, we increment the counter, and decrement it for a right edge. Each point where the counter is non-zero is covered by a rectangle. The total number of covered points is the answer.

⋆⋆★⋆⋆