2014 Rocky Mountain Regional Programming Contest

Solution Sketches
Credits

- Darko Aleksic
- Babak Behsaz
- Howard Cheng
- Peter Høyer
- Warren MacEvoy
xxx submissions, xxx correct, first correct: xxx minutes

Straightforward

Process each task in order, keeping track of the total number of minutes so far.
Eligibility

- xxx submissions, xxx correct, first correct: xxx minutes
- Straightforward
- Just apply the rules one at a time, stopping as soon as a decision is known.
- No need to look at month/day of a date, just the year.
Dynamic Programming

Let $f(n, w) =$ the maximum revenue that can be obtained when there are $n$ seats left and $w$ weeks before the flight.

Base case: $f(n, w) = 0$ when $n \leq 0$ or $w < 0$.

Recursion:

$$f(n, w) = \max_{1 \leq i \leq K_w} \{ f(n - s_{i,w}, w - 1) + p_{i,w} \cdot s_{i,w} \}$$

Minor adjustment above if $n < s_{i,w}$.

Complexity: $O(NWK)$
xxx submissions, xxx correct, first correct: xxx minutes

We can model this as a bipartite graph: one set of nodes are the clients, and the other set of nodes are the potential locations.

We connect a client to a location if the cost is 0.

The locality property implies that each connected component is a complete bipartite subgraph—so each connected component can be served by just one facility.

i.e. If the number of connected components is at most $k$, then it is possible.
Repeated Substrings

- xxx submissions, xxx correct, first correct: xxx minutes
- Use suffix arrays and the Longest Common Prefix (LCP) array.
- Whenever LCP[i] > LCP[i – 1], the difference is the number of unique substrings repeated.
- Sum up all such differences.
- For the first sample input “aabaab”

<table>
<thead>
<tr>
<th>Prefix</th>
<th>LCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>aab</td>
<td>0</td>
</tr>
<tr>
<td>aabaab</td>
<td>3</td>
</tr>
<tr>
<td>ab</td>
<td>1</td>
</tr>
<tr>
<td>abaab</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>baab</td>
<td>1</td>
</tr>
</tbody>
</table>
xxx submissions, xxx correct, first correct: xxx minutes

Form the weighted undirected graph as given.

Without any insecure buildings, this is just the standard Minimum Spanning Tree problem.

The insecure buildings must be leaves in the spanning tree, the other ones can be internal nodes or leaves.

Compute the MST without the insecure buildings. For each insecure building, connect it to the MST using the cheapest edge.
Aquarium Tank

- xxx submissions, xxx correct, first correct: xxx minutes
- Two possible approaches (among others):
  - First approach: “walk up” the polygon and figure out how high the water goes.
  - Second approach: guess the height and compute the resulting volume. Use binary search to refine the height.
- Either way: need to intersect polygon with horizontal lines, and compute the area of a polygon or trapezoid.
Restaurant Ratings

- xxx submissions, xxx correct, first correct: xxx minutes

Approach 1: make use of the fact that the number of integer solutions \(x_1, \ldots, x_n \geq 0\) such that \(x_1 + \cdots + x_n = r\) is \(\binom{r+n-1}{r}\).
  - We can find the number of ratings less than the given total.
  - Use the above to find the number of worse ratings with the same total, but with the same first \(k\) ratings

Approach 2: Dynamic Programming
  - State is \((a, k, s)\): \(a\) is rating already worse?, \(k\) rating index, \(s\) remaining rating sum.
  - \(f(1, k, s) = \sum_{i=0}^{s} f(1, k+1, s-i)\)
  - \(f(0, k, s) = f(0, k+1, s-r_k) + \sum_{i=0}^{r_k-1} f(1, k+1, s-i)\)
    \[+ \sum_{i=r_k+1}^{s} f(1, k+1, s-1-i)\]
  - base case \(f(x, n, y) = f(x, y, 0) = 1\), answer is \(f(0, 0, S)\) where \(S\) is the total rating

RMRC 2014 Solution Sketches
The answer is $\binom{n}{m-1}$.

For each subset of $m - 1$ bandits, there must be at least one lock that they cannot open (lower bound).

For each subset of $m - 1$ bandits, have one lock such that the keys are distributed to all others who are not in the subset. Any group of $m$ bandits must have a key to every lock (upper bound).
xxx submissions, xxx correct, first correct: xxx minutes

Exhaustive search.

Just try all possible $2^n$ truth-value assignment to the variables and test if the clauses are all satisfied.