Alberta Collegiate Programming Contest

2019
Good job everyone! - Judges :)

You’re all breathtaking!
Statistics
### % of teams solving a problem: estimated and actual

<table>
<thead>
<tr>
<th>Problem</th>
<th>Tony</th>
<th>Martin</th>
<th>Shaheed</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Interests</td>
<td>30%</td>
<td>25%</td>
<td>10%</td>
<td>33%</td>
</tr>
<tr>
<td>Grenade</td>
<td>20%</td>
<td>1%</td>
<td>1%</td>
<td>36%</td>
</tr>
</tbody>
</table>
| Voting                        | 5%    | 0%     | 1%      | 0% :(
| Zion’s Grocery Store          | 90%   | 90%    | 99%     | 88%    |
| Androids Frontline            | 30%   | 40%    | 40%     | 33%    |
| Hans Vick                     | 90%   | 95%    | 90%     | 82%    |
| High Table                    | 50%   | 30%    | 60%     | 56%    |
| Zencrypted Zessages           | 50%   | 65%    | 55%     | 66%    |
# Quickest Time To Solve a Problem

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Interests</td>
<td>66</td>
</tr>
<tr>
<td>Grenade</td>
<td>102</td>
</tr>
<tr>
<td>Voting</td>
<td>N/A</td>
</tr>
<tr>
<td>Zion’s Grocery Store</td>
<td>7</td>
</tr>
<tr>
<td>Androids Frontline</td>
<td>102</td>
</tr>
<tr>
<td>Hans Vick</td>
<td>6</td>
</tr>
<tr>
<td>High Table</td>
<td>82</td>
</tr>
<tr>
<td>Zencrypted Zessages</td>
<td>7</td>
</tr>
</tbody>
</table>
Solutions
Not going to go over details or take questions. Come to the next CPC meeting!
Common Interests

- Data Structures
- Author: Wenli Looi
Common Interests

- Given everyone’s interests, determine if 2 people have a common interest.
- Brute force is too slow.
- However you can show it will work if you do the following 2 things:
  - Memoize the answers: If the same query shows up again, return the cached answer.
  - When checking if 2 people have common interests, loop through the person with fewer interests and check if they are present in the other person.
Given $n$ circles, each with radius $r$, determine if it’s possible add another circle with radius $r$ that intersects with all $n$ circles.

There’s a solution iff $n$ circles with radius $2r$ overlaps at some area.

One way is to check if there exists an intersection point between any two circles which is contained within all $n$ circles.
Voting

- Combinatorics
- Author: Tony Cai
Problem is asking for number of ways to pick \( m \) subsets from a set of \( n \) elements, where each subset has size \( k \), such that the intersection of all subsets is empty.

Let \( f(i, j) \) = number of configurations such where the intersection of \( m \) subsets (each with \( i \) elements) have \( j \) elements.

\[
    f(i, j) = \text{total number of ways to choose } m \text{ subsets of size } i - (\text{ways to choose 1 intersecting element}) \cdot f(i-1, j-1) - (\text{ways to choose 2 intersecting elements}) \cdot f(i-2, j-2) - \ldots
\]

Can be solved with dynamic programming (memoizing only on \( i \), as \( j \) is implied from \( i \)).
Androids Frontline

- Graph
- Author: Martin Tran
Androids Frontline

- Basic graph traversal
- Notice that any nodes with units essentially mean that all of that node’s neighbours are directly connected.
Hans Vick

- Greedy
- Author: Martin Tran
Hans Vick

- Simple greedy problem.
- Sort by highest number of remaining rounds in magazines.
High Table

- Tree
- Author: Modan Han
High Table

- Construct tree, for all defects, traverse subtree.
- Can stop when visiting an already defected vertex. Otherwise will TLE.
Zion’s Grocery

- Implementation
- Author: Shaheed Ebrahim
Zions Grocery Store

Get the last 3 numbers of each SKU -> Add them up -> Mod by 5
Implementation
Author: Modan and Shaheed
Zencrypted Zessages

Kind of annoying, but it’s just implementation!

Shouldn’t construct new strings every every split/concat, will TLE. Instead read/print from string in place, or use better things like StringBuilder in Java or `.join()` in Python.

Also shouldn’t linearly search for next start point, will TLE. Sort input pairs or similar.
Thanks to everyone that participated, and thanks to everyone who helped working on this contest!