ACPC 2020

Solutions For All Problems
Nice job everyone! S/O to our Judges and Problem setters :) You’re all breathtaking!
Thank you to Arcurve for sponsoring this event!
Placements

Note: These are not confirmed, we still have to check over the submissions. For plagiarism, validity, etc.
Division 2
Second Place:
Leo Gao
Division 2
First Place:
Bad GNUs
Division 1

Third Place:

:person_running: :person_running: :person_running:
Division 1
Second Place:
Benyamin Bashari
Division 1
First Place:
Praxis Makes Perfect
<table>
<thead>
<tr>
<th>Problems:</th>
<th>Predicted % (from problem setters)</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Division 1</td>
</tr>
<tr>
<td>Best Investing</td>
<td>70%</td>
<td>4%</td>
</tr>
<tr>
<td>Hired Help</td>
<td>30%</td>
<td>40%</td>
</tr>
<tr>
<td>Laptop Stickers</td>
<td>70%</td>
<td>88%</td>
</tr>
<tr>
<td>Listen To Your Boss</td>
<td>30%</td>
<td>28%</td>
</tr>
<tr>
<td>Monochromatic Minesweeper</td>
<td>10%</td>
<td>4%</td>
</tr>
<tr>
<td>Password Rotation</td>
<td>10%</td>
<td>16%</td>
</tr>
<tr>
<td>Path Crossings</td>
<td>60%</td>
<td>56%</td>
</tr>
<tr>
<td>Straights</td>
<td>70%</td>
<td>96%</td>
</tr>
<tr>
<td>Test Drive</td>
<td>95%</td>
<td>X</td>
</tr>
<tr>
<td>Warring Scoring</td>
<td>95%</td>
<td>X</td>
</tr>
<tr>
<td>Wormholes Extreme!!!</td>
<td>40%</td>
<td>24%</td>
</tr>
<tr>
<td>Wrapping Trees</td>
<td>10%</td>
<td>4%</td>
</tr>
</tbody>
</table>
## Quickest Time to Solve a Problem

*\(\times\) = not part of set

<table>
<thead>
<tr>
<th>Problems</th>
<th>Division 1</th>
<th>Division 2</th>
<th>Open Division</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Investing</td>
<td>171</td>
<td>184</td>
<td>135</td>
</tr>
<tr>
<td>Hired Help</td>
<td>34</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td>Laptop Stickers</td>
<td>3</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>Listen To Your Boss</td>
<td>83</td>
<td>186</td>
<td>10</td>
</tr>
<tr>
<td>Monochromatic Minesweeper</td>
<td>277</td>
<td>√</td>
<td>137</td>
</tr>
<tr>
<td>Password Rotation</td>
<td>53</td>
<td>251</td>
<td>20</td>
</tr>
<tr>
<td>Path Crossings</td>
<td>53</td>
<td>229</td>
<td>10</td>
</tr>
<tr>
<td>Straights</td>
<td>14</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>Test Drive</td>
<td>√</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>Warring Scoring</td>
<td>√</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>Wormholes Extreme!!!</td>
<td>143</td>
<td>N/A</td>
<td>43</td>
</tr>
<tr>
<td>Wrapping Trees</td>
<td>286</td>
<td>√</td>
<td>20</td>
</tr>
</tbody>
</table>
BEST INVESTING

• Within a loop:
  • Work backwards from end year withdrawing maximum amount
  • Work forward from zero years depositing the maximum amount

• When the years are equal, you’ll have the maximum net gain

• Compound Interest
• Dante Bencivenga
HIRED HELP

• Sort all the deadlines in ascending order
• Use binary search to compute the solution OR solve in linear-time

• Greedy
• Zac Friggstad
LAPTOP STICKERS

• Painter’s algorithm
  • Creates images by sorting the stickers within the image by their depth and placing each stickers in order from the farthest to the closest object.

• Ad-hoc
• Martin Tran
LISTEN TO YOUR BOSS

- Lowest Common Ancestor (LCA) in a tree algorithm

- Graph
- Wenli Looi
MONOCHROMATIC MINESWEEPER

- Brute force along smaller dimensions and bitset DP along larger dimensions
- Pruning of invalid partial solutions is required
PASSWORD ROTATION

- Lexicographically minimal string rotation
  - Booth’s algorithm

- String
  - Wenli Looi
PATH CROSSINGS

• Check all pairs
  • Use sliding window technique to look at points in 10 second time window

• Geometry, ad-hoc
• Wenli Looi
STRAIGHTS

- Put inputs into an array
- Sort array
- Add array elements into hash map adding 1 every time subtract 1 if deemed necessary
- Add sum of Hash Map

- Greedy, hash map
- Jonathan Chong
TEST DRIVE

• Easy input comparisons

• Ad-hoc
• Dante Bencivenga
WARRING SCORE

- Check score system and compare results

- Ad-hoc
- Dante Bencivenga
WORMHOLE EXTREME!!!

• Find minimum initial velocity for each segment in reverse order
  • Can use binary search

• Math, iterative
• Zac Friggstad
WRAPPING TREES

- For each '1' pixel at coordinates \((x,y)\) add the point \((x/(n-1)*n, y/(n-1)*n)\) to the convex hull

- At the last step, compute the perimeter of the convex hull
Conclusion

- We will be going over some of the problems IN-DEPTH on our next weekly Wednesday lecture.
- There will be a poll on discord for what problems you would like us to go over.
- If you haven't already join our discord it's our main source of communication:
  - https://discord.com/invite/MEXwfze