T183 - Exam Anti-Cheat

Percy Wong {percywtc}
The Problem

Given a set of $N$ points $(x[i], y[i])$

- $0 \leq x[i], y[i] \leq 1000$
- $x[i], y[i]$ are integers

Using $V$ colors to color the points

- such that the closest pair of points having same color
- distance is maximized

Output-only task

- $10^{(M-D)/(T-D)}$ points for each of the 10 test cases
### Statistics

<table>
<thead>
<tr>
<th>Max</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Subtasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>88.660</td>
<td>33.327</td>
<td>21.322</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:0</td>
<td>7.328:1</td>
<td>3.680:1</td>
<td>10:0:1</td>
</tr>
<tr>
<td>5.085:3</td>
<td>8.861:1</td>
<td>5.136:1</td>
<td>4.179:1</td>
</tr>
<tr>
<td>4.269:2</td>
<td>6.655:1</td>
<td>4.366:1</td>
<td>2.673:1</td>
</tr>
<tr>
<td>1.933:1</td>
<td>6.151:1</td>
<td>2.525:1</td>
<td>6.937:1</td>
</tr>
<tr>
<td>1.637:1</td>
<td>5.303:1</td>
<td>1.338:1</td>
<td>5.233:1</td>
</tr>
<tr>
<td>1.391:1</td>
<td>3.319:1</td>
<td>1.221:1</td>
<td>4.521:1</td>
</tr>
<tr>
<td>1.358:1</td>
<td>3.101:1</td>
<td>1.127:1</td>
<td>1.246:1</td>
</tr>
<tr>
<td>1.24:1</td>
<td>2.315:3</td>
<td>1.17:2</td>
<td></td>
</tr>
<tr>
<td>1.132:2</td>
<td>1.866:1</td>
<td>1.074:4</td>
<td></td>
</tr>
<tr>
<td>1.061:17</td>
<td>1.587:18</td>
<td>1.064:2</td>
<td></td>
</tr>
<tr>
<td>1:1</td>
<td>1.054:2</td>
<td>1:33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.296:3</td>
<td>1:295:1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.135:1</td>
<td>1:4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1:12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case #1

\[ N = 62 \]
\[ V = 2 \]
\[ T = 53.7 \]
\[ D = 37.000 \]

Trainer’s best = 53.759
Case #2

N = 100
V = 2
T = 124.0
D = 5.000
Trainer’s best = 124.004
Case #3

N = 123
V = 2
T = 32.0
D = 30.529
Trainer’s best = 32.016
Case #4

N = 942
V = 2
T = 4.1
D = 3.162
Trainer’s best = 4.123
Case #5

N = 777
V = 3
T = 19.2
D = 14.765
Trainer’s best = 19.209
Case #6

N = 256
V = 3
T = 33.3
D = 1.414
Trainer’s best = 33.302
Case #7

N = 512
V = 4
T = 77.0
D = 1.000
Trainer’s best = 77.000
Case #8

N = 947
V = 4
T = 22.8
D = 12.728
Trainer’s best = 22.825
Case #9

N = 999
V = 5
T = 60.8
D = 1.000
Trainer’s best = 60.803
Case #10

N = 1000
V = 5
T = 55.1
D = 25.807
Trainer’s best = 55.145
Solution for 10 points

- $10^{(M-D)/(T-D)}$ points for each of the 10 test cases
- meaning if you get $M = D$, you score 1 point in each test case
- as $D$ is the minimum distance of a pair in the input
- outputting AAAAAAA... can obtain $T = D$
- easy 10 points :)

\[ \text{M-D)}/(T-D) \]
Solution for ~10.5 points

- randomly assign colors to the points

Solution for ~12 points

- outputing the characters periodically, i.e. ABCDABCDABCD...

Solution for ~22 points

- randomly assign colors to the points
- repeat this 10000 times, output the best one
Solution for ~48 points

- you should know that for \( V = 2 \), there exists optimal solution
- we can do it by *binary search on answer*
- building edges between points with \( \text{dist} < \text{mid} \)
- check if the graph is bipartite or not

Solution for ~50 points

- for \( V > 2 \), divide the points into two groups
- for each group, find the optimal solution with the above method
- repeat this 10000 times, output the best one
Solution for ~65 points

- randomly assign colors to the points first
- do the following 10000 times:
  - find the pair of same color with minimum distance
  - randomly select one of them and change its color randomly
  - if any of them is selected in the previous round, must select the other one
Solution for ~72 points

- randomly assign colors to the points first
- do the following many times:
  - find all pairs of same color with minimum distance
  - for all points in these pairs, randomly assign colors
  - stop when no better results obtained
Solution for ~90 points

- binary search on answer (you can do it manually too)
- build the edges with \( \text{dist} \leq \text{mid} \)
- now your task is to color the graph
- instead of exhausting the colors for all nodes
- only exhaust nodes with degree \( \geq V \)
  - as all nodes with degree \(< V\) can be colored easily after knowing neighbours’ colors
  - one of the techniques in S141 - Dividing the Cities
- if you cannot find any solutions / too slow because of too many nodes
- reduce the upperbound
Solution for ~99.6 points

- randomly assign colors to nodes first
- do the following 6000 times:
  - randomly choose two different colors
  - for all points currently assigned with these two colors
  - run the “optimal solution” again to reassign their colors
  - *you should randomly choose the colors for each component*
  - *so that the number of nodes with different colors are balanced*
Techniques

- The points of the mentioned solutions are calculated by
  - just running that single solution to solve all the inputs
  - in contest, try not to 「一 CODE 走天涯」
  - you may actually get more points by comparing different solutions
    - easily done with *cumulative scoring* :)
- Observe patterns in the input data
  - Case #1, #3 and #4
Techniques

• Try to come up with as many ideas as possible, some ideas may work better than you think
  ○ Could be weak inputs, weak scoring function or inherent limitation of the problem
  ○ Assess which idea is the most cost-effective (cost = coding time)

• Visualization
  ○ If provided, you can use a spreadsheet program to make charts
  ○ Write a program to generate svg graphics and view them via a web browser

(from T174 - Constellation solution PPT)