NWERC 2018 presentation of solutions

NWERC 2018 Jury

- **Per Austrin**
  KTH Royal Institute of Technology
- **Gregor Behnke**
  Ulm University
- **Jeroen Bransen**
  Chordify
- **Alexander Dietsch**
  FAU Erlangen-Nürnberg
- **Arthur van Goethem**
  Eindhoven University of Technology
- **Bjarki Ágúst Guðmundsson**
  Syndis
- **Irina Kostitsyna**
  Eindhoven University of Technology
- **Stefan Kraus**
  FAU Erlangen-Nürnberg
- **Robin Lee**
  Google
- **Simon Lindholm**
  KTH Royal Institute of Technology
- **Lukáš Poláček**
  Innovatrics
- **Alexander Raß**
  FAU Erlangen-Nürnberg
- **Philipp Reger**
  FAU Erlangen-Nürnberg
- **Tobias Werth**
  Google
- **Paul Wild**
  FAU Erlangen-Nürnberg

Big thanks to our test solvers

- **Eduard Kalinicenko**
  Palantir
- **Timon Knigge**
  Google
- **Ragnar Groot Koerkamp**
  Google
- **Jan Kuipers**
  bol.com
- **Dominik Paulus**
  Google
- **Tobias Polzer**
  Google

I: Inflation

Problem Author: Arthur van Goethem

Problem

Assign gas canisters to balloons to maximize $\min(\frac{c_i}{i})$, such that $\forall i. \frac{c_i}{i} \leq 1$

Solution

1. Sort c
2. If $\exists i. \frac{c_i}{i} > 1$, print `impossible` and return
3. Print $\min(\frac{c_i}{i})$

Statistics: 146 submissions, 118 accepted

I: Inflation

Problem Author: Arthur van Goethem

Jury: behind the scenes

`--- November 17, 2018 ---`

- **Robin**: Good, we're all set for the contest.
- *Per has entered the room*
- *Per has successfully challenged* `jeroen.java`
- *Per has successfully challenged* `rgl.java`
- *Per has successfully challenged* `tobi.kt`
- **Jeroen**: What?!
- **Per**: They all use `*CENSORED*` which is `*CENSORED*`.
- **Simon**: Let's not include those cases.
- **Others**: Indeed, because we are so nice.

K: Kleptography

Problem Author: Alexander Dietsch

Problem

Given ciphertext $b_1 b_2 \cdots b_m$ encrypted with the Autokey cipher and last n letters $a_{m-n+1} \cdots a_m$ of plaintext, recover entire plaintext a.

Autokey cipher recap:

- Key $k_1 k_2 \cdots k_n$, gets padded with plaintext ($k_{n+i} = a_i$).
- Plaintext a encrypted by $b_i = (a_i + k_i) \bmod 26$.

Solution

1. For all $i \leq m - n$, we have $a_i = k_{i+n} = (b_{i+n} - a_{i+n}) \bmod 26$.
2. Compute for i from $m - n$ down to 1.
3. Complexity $O(m)$.

Statistics: 126 submissions, 117 accepted

H: Hard Drive

Problem Author: Stefan Kraus

Problem

Create a bitstring of length n such that:

- The number of bit changes is c
- The bits at given positions z_i are all 0

Solution

1. If c is even, start with a 0, otherwise a 1
2. Greedily alternate 0 and 1 where possible, as long as changes are needed

Statistics: 227 submissions, 117 accepted

B: Brexit Negotiations

Problem Author: Gregor Behnke

Problem

Given a DAG with vertex weights $e(i)$, find a topological ordering π that minimizes

$$\max_i(e(\pi(i)) + i).$$

Solution 1 more helpfully

- Basic idea: last vertex should have as small $e(i)$ as possible.
- Adapt standard topological sort.
  - Build schedule from end.
  - Keep adding currently available topic with smallest $e(i)$.
  - Maintain available topics in priority queue.
- Time complexity $O(n \log n + m)$.

B: Brexit Negotiations

Problem Author: Gregor Behnke

Solution 2

- Ideally would like to sort vertices by decreasing value of $e(i)$, but that might violate the precedence constraints.
- Assign potential $p(i) = -e(i)$ for each vertex.
- Propagate potentials: for each predecessor j of i, must have $p(j) \leq p(i) - 1$.
- Sorting by propagated potentials gives a valid and optimal ordering.

Statistics: 212 submissions, 72 accepted

B: Brexit Negotiations

Problem Author: Gregor Behnke

UK's Brexit deal agreed by EU leaders

Theresa May says she agrees with EU officials that this "is the best and only deal possible".

UK

Credit: BBC.com

G: Game Design

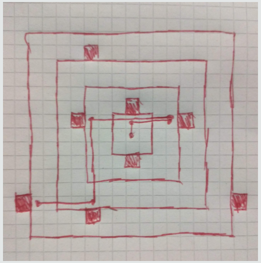
Problem Author: Alexander Dietsch, Robin Lee & Tobias Werth

Problem

Given a sequence of UP/DOWN/LEFT/RIGHT instructions, construct a sliding ball maze with the given instructions as a solution.

Solution

1. Start at centre
2. Each time we turn ± 90 degrees, extend bounding box of maze by 2, and add blocks in both directions at edge of bounding box.
3. At end, shift maze so that we end at $(0, 0)$ rather than starting there.
4. Time complexity $O(n)$.



G: Game Design

Problem Author: Alexander Dietsch, Robin Lee & Tobias Werth

Solution 2

1. Brute force: for each instruction try making it 1 step, 2 steps, etc and recursively solve the rest.
2. To make it fast enough, good to figure out when answer is `impossible`. This happens if and only if input ends with LRL, RLR, or UDU, DUD.

Solution 3

1. Randomly place blocks (make each cell a block with probability 5%).
2. Run walk from $(0, 0)$, if all steps can be executed and we end up in a position not visited earlier in the walk then this gives a solution.

Statistics: 121 submissions, 44 accepted

J: Jinxed Betting

Problem Author: Alexander Raß

Problem

Betting tournament:

- Each round we bet between two options, each correct bet gets a point.
- Julia starts in the lead, has terrible luck but compensates by copying the majority bet from the runners-up.
- For how many rounds will she stay in the lead?

(Not a) Solution

1. In a round with t runners-ups, worst thing that can happen is that their bets are split evenly $\lceil t/2 \rceil$.
2. Naive simulation: $\Theta(r \cdot n)$ where r is number of rounds, which can be as large as 10^{16}.
3. Mildly better simulation: keep track of Julia's lead over the others instead of their scores. $\Theta(r)$ time instead, still a year or so too slow.

J: Jinxed Betting

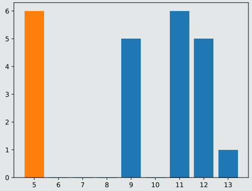
Problem Author: Alexander Raß

Further Analysis

x axis: number of points behind J.

y axis: number of bettors

leftmost bar: group of runners-up



1. In next $1 + \lfloor \log_2 t \rfloor$ rounds, each of the t runners-up catch up in all but one round, remaining bettors catch up in all rounds.
2. If the t runners-up initially have a lead of d over the next group of bettors, this pattern repeats d times, then the two groups are joined and the number of runners-up grows.
3. Keep running this sped-up simulation until J. no longer in the lead.
4. Complexity $O(n \log n)$ for sorting the scores, then $O(n)$ arithmetic operations.

Statistics: 231 submissions, 34 accepted

C: Circuit Design

Problem Author: Paul Wild

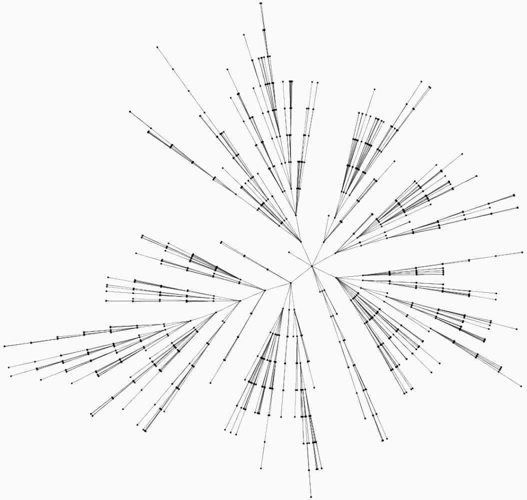
Problem

Given a tree, assign each point a position on a plane such that:

- The distance between any two connected points is (approx.) 1
- Points are not too close together
- Edges do not intersect
- $-3000 \leq x, y \leq 3000$

C: Circuit Design

Problem Author: Paul Wild



C: Circuit Design

Problem Author: Paul Wild

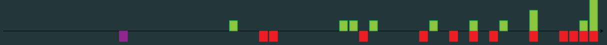
Solution

1. Pick arbitrary root and place at origin
2. Draw points in region between two angles
3. Split region into subregions for children proportional to their width
4. Alternative: Instead of angle, use x-coordinates between -1 and 1, and increase y

Pitfalls

1. Print enough digits (9 could be too few)

Statistics: 190 submissions, 48 accepted



A: Access Points

Problem Author: Arthur van Goethem

Problem

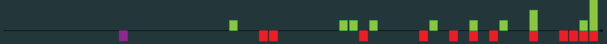
Given list of n points $p_1, \ldots, p_n$ in $\mathbb{R}^2$, find n points $q_1, \ldots, q_n$ such that q_i is componentwise smaller than q_{i+1}, and $\sum_{i=1}^{n} \|q_i - p_i\|_2^2$ is minimized.

Solution

- $\|q_i - p_i\|_2^2 = (x(p_i) - x(q_i))^2 + (y(p_i) - y(q_i))^2$
- x and y coordinates do not interact, solve them separately and add up the answers.
- Problem reformulated:
  given sequence $a_1, \ldots, a_n \in \mathbb{R}$, find $x_1 \leq x_2 \leq \ldots \leq x_n$ such that $\sum (x_i - a_i)^2$ is minimized.

A: Access Points

Problem Author: Arthur van Goethem



The 1D case

- Add item by item.
- When adding x_i, it wants to go to position a_i.
- If a_i smaller than position of previous item, x_i ends up in same position as previous item and pushes it (and possibly more items) towards the left.
- View x_i together with previous item as a new "meta-item".
  (Previous item may already have been a meta-item, so these can get larger and larger.)
- Where does meta-item consisting of items $x_j, \ldots, x_i$ want to go?
  $\sum_{k=j}^{i}(x - a_k)^2$ is minimized by $x = \text{avg}(a_j, \ldots, a_i) = \frac{1}{i-j+1}\sum_{k=j}^{i} a_k$.
- Keep number of items and $\sum a_k$ for each meta-item to quickly be able to add more things to it.
- While positions of the last two meta-items out of order, merge them.
- Time complexity $O(n)$.

Statistics: 26 submissions, 13 accepted

E: Equality Control

Problem Author: Robin Lee

Problem

Consider a programming language whose expressions consist of

- elementary lists: $[x_1, \ldots, x_n]$
- concating two lists: $concat(E_1, E_2)$
- random reordering: $shuffle(E)$
- sorting: $sorted(E)$

Question: Given two expressions, are the distributions the same?

(Actual) Problems

Explicit computation of the distribution is not possible due to size.

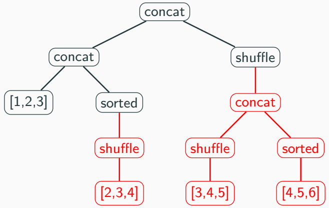
Monte-Carlo sampling also does not work.

E: Equality Control

Problem Author: Robin Lee

Algebra

- Operations below *sorted* or *shuffle* are irrelevant!
- Compact expressions to sequences of lists (sorts can be executed directly) and *shuffle*s.
- Expressions are identical iff sequences are



Pitfalls

- $\textit{shuffle}([1, 1, 1]) = [1, 1, 1]$
- Adjacent lists must be joined

E: Equality Control

Problem Author: Robin Lee

Sampling - possibility 2

Take two samples per expression, i.e., $s_1^{E_1}, s_2^{E_1}$ and $s_1^{E_2}, s_2^{E_2}$ with:

1. s_1 : $\mathit{shuffle} = \mathit{sorted}$
2. s_2 : $\mathit{shuffle} = \mathit{reverse} \circ \mathit{sorted}$

If $s_1^{E_1} = s_1^{E_2}$ and $s_2^{E_1} = s_2^{E_2}$ then `equal` else `not equal`.

Statistics: 127 submissions, 20 accepted

D: Date Pickup

Problem Author: Bjarki Ágúst Guðmundsson

Problem

Given a directed graph, find a walk from vertex s minimizing the maximum shortest distance to vertex t during time $[a, b]$.

Solution

1. Distances will be needed... compute distance from s to all vertices, and from all vertices to t, using two runs of Dijkstra's algorithm.
2. Binary search on the answer.
3. Now just need to check if given delay δ is possible.

D: Date Pickup

Problem Author: Bjarki Ágúst Guðmundsson

Checking if delay $\leq \delta$ possible

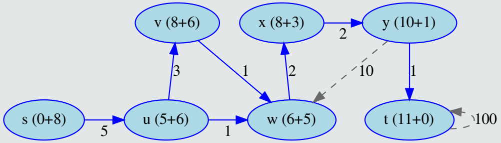
1. Mark vertices u as "good" if $d(s, u) + d(u, t) \leq a + \delta$.
   (If u does not satisfy this, then any route through u will give delay $> \delta$ if signal comes at time a.)
2. Propagate: if u is good and $u \xrightarrow{\ell} v$ with $\ell + d(v, t) \leq \delta$, then mark v and edge as good too.
   (For such edges we get delay $\leq \delta$ if signal comes when traversing the edge.)
3. If subgraph of good edges has a cycle: delay δ is possible.
   (Can just cycle around indefinitely until signal comes.)
4. Otherwise, subgraph is a DAG. Use dynprog to compute longest time we can stay in the subgraph.
   If this is $\geq b$, delay δ is possible.
5. Complexity $O(n)$ for the check.

D: Date Pickup

Problem Author: Bjarki Ágúst Guðmundsson

Checking if delay $\leq \delta$ possible, example

- $\delta = 10$:
  1. vertex v not initially good ($d(s,v) + d(v,t) = 8 + 6 > a + \delta$).
  2. propagate: edge $u \to v$ and v marked as good.
  3. no cycle, compute longest paths
  4. can stay at s until time $a + \delta - d(s,t) = 4$
  5. propagating $\implies$ can arrive at t at time 18 at the latest
  6. delay 10 *not possible* (but would be possible if $b \leq 18$)



$a = 2, b = 20.$

Legend: "u ($d(s,u) + d(u,t)$)"

D: Date Pickup

Problem Author: Bjarki Ágúst Guðmundsson

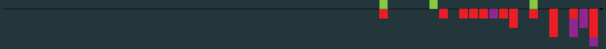
Checking if delay $\leq \delta$ possible

1. Mark vertices u as "good" if $d(s, u) + d(u, t) \leq a + \delta$.
   (If u does not satisfy this, then any route through u will give delay $> \delta$ if signal comes at time a.)
2. Propagate: if u is good and $u \xrightarrow{\ell} v$ with $\ell + d(v, t) \leq \delta$, then mark v and edge as good too.
   (For such edges we get delay $\leq \delta$ if signal comes when traversing the edge.)
3. If subgraph of good edges has a cycle: delay δ is possible.
   (Can just cycle around indefinitely until signal comes.)
4. Otherwise, subgraph is a DAG. Use dynprog to compute longest time we can stay in the subgraph. If this is $\geq b$, delay δ is possible.
5. Complexity $O(n)$ for the check.

Statistics: 13 submissions, 1 accepted

F: Fastest Speedrun

Problem Author: Simon Lindholm



Problem

Beat a game as fast as possibly by solving the levels in an optimal order using special items.

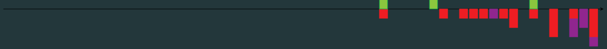
Solution

General idea: smart preprocessing + DP

1. Each level should be solved, so add s_i to answer and subtract from all $a_{i,j}$
2. Now special item can always be used for free
3. The special items form a set of cycles with trees pointed at them
4. For each cycle, find the cheapest $a_{i,n}$, add it to the answer and subtract from cycle
5. Now if we get the highest item, we can solve all the rest for free
6. Now use DP (or Dijkstra) to find the cheapest way of obtaining that
7. $O(n^2)$

F: Fastest Speedrun

Problem Author: Simon Lindholm

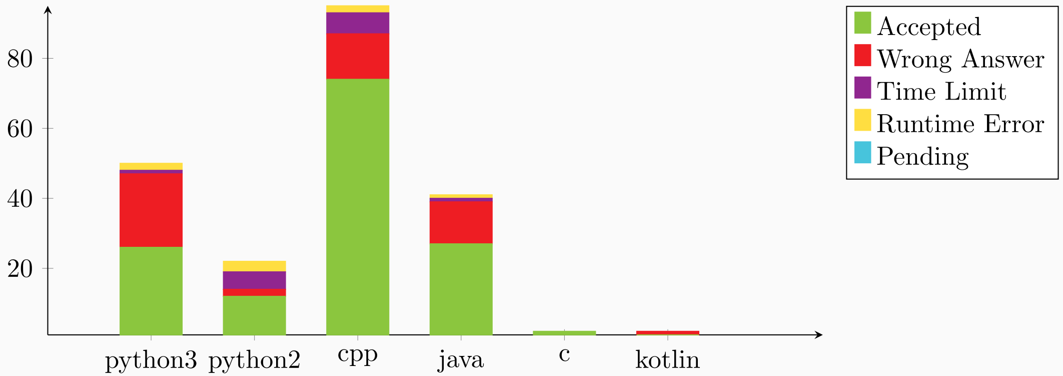


Solution 2

- Problem can be reduced to *Minimum directed spanning tree*
- Use algorithm from TCR (Edmonds' algorithm)
- Make sure it is bug-free and runs in $O(n^2)$

Statistics: 25 submissions, 3 accepted

Language stats


python3
python2
cpp
java
c
kotlin
🟩 Accepted
🟥 Wrong Answer
🟪 Time Limit
🟨 Runtime Error
🟦 Pending