GCPC 2026 Presentation of Solutions

German Collegiate Programming Contest 2026

The GCPC Jury

June 12, 2026

GCPC 2026 Jury

- **Andreas Grigorjew**
  Paris Dauphine FR, CPUlm
- **Konrad Letz**
  Hasso-Plattner-Institut, Potsdam
- **Niklas Mohrin**
  Hasso-Plattner-Institut, Potsdam
- **Jannik Olbrich**
  Ulm University, CPUlm
- **Erik Sünderhauf**
  Technical University of Munich
- **Lucas Schwebler**
  Karlsruhe Institute of Technology
- **Leonard Weininger**
  RWTH Aachen
- **Christopher Weyand**
  MOIA GmbH, CPUlm
- **Yidi Zang**
  Karlsruhe Institute of Technology, CPUlm
- **Michael Zündorf**
  Karlsruhe Institute of Technology, CPUlm

GCPC 2026 Test Solvers

- **Michael Ruderer**
  University of Augsburg, CPUlm
- **Paul Wild**
  Friedrich-Alexander University Erlangen-Nürnberg, CPUlm
- **Wendy Yi**
  Karlsruhe Institute of Technology, CPUlm

GCPC 2026 Organization and Technical Team

- **Felicia Lucke**
  ENS Lyon FR, CPUlm
- **Nathan Maier**
  CPUlm
- **Alexander Schmid**
  CPUlm
- **Pascal Weber**
  University of Vienna, CPUlm

A: Attracting Attendees

Problem author: Leonard Weininger

Problem

Given n bands and m fans and connections between them, select some of them such that:

- each band is connected to at least c fans
- each fan is connected to at least halve of the bands he was initially connected to

Solution

- initially consider all bands and fans
- if a band has less than c fans left you must discard it
- if a fan has less than halve of its bands remaining you must discard him
- if nothing must be discarded the remaining bands and fans are a valid solution

  (this is actually the largest possible solution)

B: Bye Bye Bilbo

Problem author: Jannik Olbrich

Problem

Given a tree, find a minimal subset of nodes, such that every node has distance less than k to this subset.

- Solve the problem greedily. Consider a node with maximum depth u.
- Walk up $k-1$ steps to the node v and add v to the subset.
- Remove the subtree rooted at v and repeat all the steps.

C: Crosses and Circles

Problem author: Konrad Letz

- **Problem:** Win tic-tac-toe on a large (59×42) grid when going first.
- Play the first move somewhere with enough space around it, for example near the center.
- After the judge's first move, choose a direction such that the adjacent cell and both completion cells are still free.
- Play the adjacent cell in that direction.
- The two possible completion cells are now two different winning threats. The judge can block at most one, so play the other one.

D: Delphi Danger

Problem author: Konrad Letz

Problem

Find n infinite binary strings s_i, which satisfy prophecies of the form

- $s_u[t] \neq s_v[t]$
- t is the smallest index with the above property

Minimize the number of 1's in those strings.

- Iterate over t in decreasing order and process all prophecies with the current t
- The restrictions $s_u[t] \neq s_v[t]$ can be satisfied iff the graph formed by the edges (u, v) is bipartite
- Greedily assign a 1 to the smaller part of each component.
- Strings have to be identical for all further prophecies $\rightarrow$ merge each connected components into a single node
- Complexity: $\mathcal{O}(m \log m + n)$ for sorting and checking bipartiteness

E: Egocentric Expedition

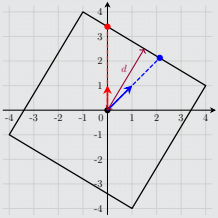
Problem author: Michael Zündorf

Problem

- Find the area of a hidden arbitrarily rotated square.
- Ask two query directions and receive the distances from square center to edge.

Solution

- Query two directions 45 degrees apart, i.e. $(1, 0)$ and $(1, 1)$.
- Assume the two points in those directions lie on one side of the square.
- Calculate minimum distance d to that side (line), the answer is $(2d)^2$.
- Why is this correct?

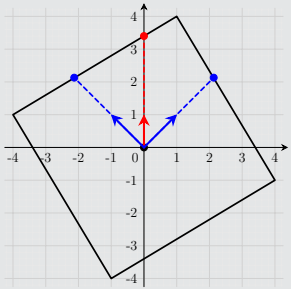


E: Egocentric Expedition

Problem author: Michael Zündorf

Solution

- Why is this correct?
- Note that a square has 90 degree rotation symmetry.
- Specifically, direction $(-1, 1)$ gives the same distance as $(1, 1)$.
- Notice that either $(0, 0)$ and $(1, 1)$ lie on the same side, or $(0, 0)$ and $(-1, 1)$.
- So assuming $(0, 0)$ and $(1, 1)$ on same side results in correct area.



E: Egocentric Expedition

Problem author: Michael Zündorf

Wrong Solution

- Just using two very close directions does not work, with given constraints, there can always be a corner in between.

F: Fighting Fraud

Problem author: Christopher Weyand

- **Problem:** Given a list of tasks that either pick up or drop off an item. Determine if each item is picked up and later dropped off exactly once.
- Group the tasks by item.
- Check that each group consists of exactly two tasks, a pickup followed by a drop-off.

G: Garbled Garden

Problem author: Lucas Schwebler, Yidi Zang

Problem

Given an array $t_1, \ldots, t_n$, sort it by applying the minimum number of permutations which have only one cycle of length > 1.

Solution

- Already sorted? $\rightarrow$ answer 0
- Check if one operation is sufficient:
  - Let s be the sorted version of t.
  - Build the graph G with directed edges $s_i \rightarrow t_i$.
  - G has multiple components (with > 0 edges)? $\rightarrow$ one operation is not sufficient.
  - Otherwise $\rightarrow$ one operation suffices, construct Euler cycle
- Two operations are always sufficient!

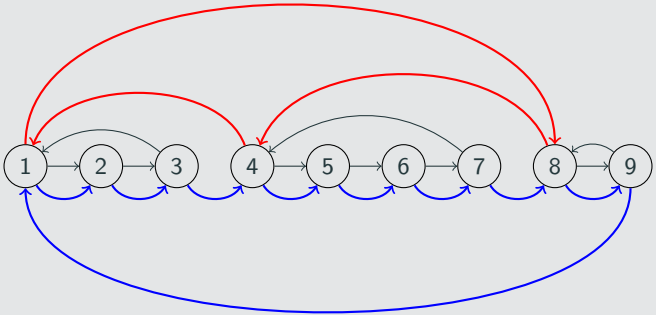
G: Garbled Garden

Problem author: Lucas Schwebler, Yidi Zang

Lemma

Every permutation is the composition of two cycles.

Proof by picture



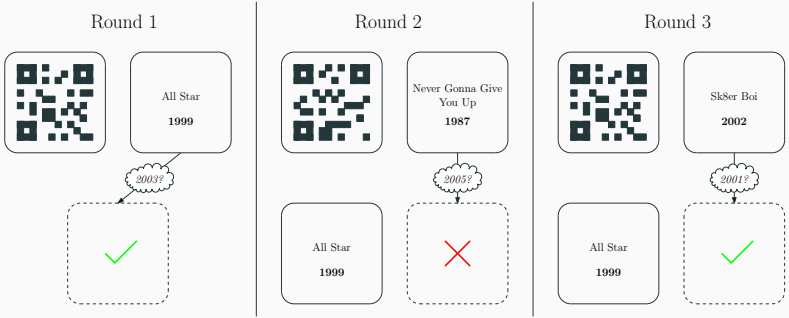
H: Historical Hits

Problem author: Lucas Schwebler

Problem

Find the expected number of correct guesses in a single player game of Hitster (random card order).

- $n \leq 3000$ songs, release year a_i, guess b_i
- a_i are distinct!



H: Historical Hits

Problem author: Lucas Schwebler

Solution

- Suppose that card i is drawn first.
- Cards j with a_i between a_j and b_j are guessed incorrectly.
- Two independent subproblems: cards with $a_j, b_j < a_i$ and cards with $a_j, b_j > a_i$.

$\rightsquigarrow$ $\mathrm{dp}(\ell, r)$ = expected value with cards $a_i, b_i \in (\ell, r)$.

- Computation (where $S(\ell, r) = \{i \mid a_i, b_i \in (\ell, r)\}$):

$$\mathrm{dp}(\ell, r) = \frac{1}{|S(\ell, r)|} \sum_{i \in S(\ell, r)} 1 + \mathrm{dp}(\ell, a_i) + \mathrm{dp}(a_i, r)$$

- $O(n^3) \rightsquigarrow$ too slow
- Optimize to $O(n^2)$ using prefix sums / sweepline techniques.
- For fixed ℓ and increasing r maintain the sum $\sum_{i \in S(\ell, r)} \mathrm{dp}(\ell, a_i)$.
- Challenge: Solve the problem with duplicate a_i allowed.

I: Incremented Itinerary

Problem author: Niklas Mohrin

Problem

Given an undirected graph. Is there a path from vertex 1 to vertex n that is exactly one edge longer than a shortest path?

Solution 1

- Let $D = \mathrm{dist}(1, n)$.
- **Observation:** Any path from 1 to n of length $D + 1$ must contain an edge $\{u, v\}$ such that the prefix from 1 to u and the suffix from v to n are shortest paths (possibly $u = 1$ or $v = n$).
- Otherwise, the path would have length at least $D + 2$.
- Therefore it suffices to check whether there exists an edge $\{u, v\}$ with

$$\mathrm{dist}(1, u) + 1 + \mathrm{dist}(v, n) = D + 1.$$

- To check this efficiently, precompute all distances from 1 and from n using two BFSs.
- Time complexity: $\mathcal{O}(n + m)$.

I: Incremented Itinerary

Problem author: Niklas Mohrin

Problem

Given an undirected graph. Is there a path from vertex 1 to vertex n that is exactly one edge longer than a shortest path?

Solution 2

- Use two states $(v, 0)$ and $(v, 1)$ per vertex.
- States (u, i) and (v, j) are connected iff there is an edge $\{u, v\}$.
- Run a BFS on these states starting from $(1, 0)$, with the following restriction:
  - States $(v, 1)$ can only be visited if reached at distance exactly one longer than the distance to $(v, 0)$.
- Finally, check if $(n, 1)$ is visited.
- Time complexity: $\mathcal{O}(n + m)$.

J: Junior Joining

Problem author: Yidi Zang

Problem

- Given $2n$ recruits with attack a_i, defense d_i, and home city c_i.
- Assign them into n pairs, maximize sum over fighting power.
- Fighting power of a pair (i, j) is $\begin{cases} a_i + d_j, & \text{if } c_i \neq c_j, \\ a_i + d_j + c_i, & \text{if } c_i = c_j \end{cases}$

Solution

- Instead of pairing, partition recruits into two groups of n, attacking and defending.
- Without city bonus, it is optimal to sort by $a_i - d_i$ and split in half.
- Therefore, for all recruits of one city, sort by $a_i - d_i$.
- Initially, split each city in half to maximize city bonus.
- If there is an odd number of recruits, ignore the middle one.
- Use middle ones to fill both groups to n.
- For each recruit, calculate attack-defense difference to move to other group.

J: Junior Joining

Problem author: Yidi Zang

Proof Sketch

- We start with optimal city bonus.
- Whenever we move one (not middle), we lose one city bonus pairing.
- If a recruit from a city c moves from group A to B, no one from c will move from B to A.
- The middle always moves before any other from their city moves.
- Notice that for each city, the recruits are always at an optimal assignment (no swap is better).
- If a better optimal solution were to exist, at least two of different cities are swapped.
- At the end of our process, no two such people can exist.

K: Keeping Cows

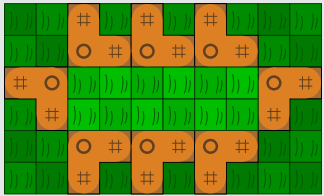
Problem author: Yidi Zang

Problem

- Place corner-fences into a 100×100 grid.
- Create a connected fenced area of exactly $1 \leq a \leq 5000$ cells.

Solution

- There are many different constructions of varying implementation difficulty.
- Determine minimum h with $(h-1) \cdot 80 \geq a$ (one extra row).
- Fence a $h \times 80$ rectangle area.

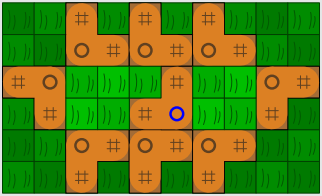


K: Keeping Cows

Problem author: Yidi Zang

Case $h = 2$

- If $h = 2$, then $a \leq 80$, and place a (blue) block as shown below.
- Notice that left component is odd, right is even.
- Move middle block left or right to increase the relevant component by 2.
- All $a \leq 80$ are achievable this way.

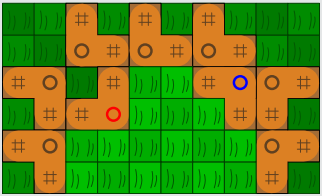


K: Keeping Cows

Problem author: Yidi Zang

Case $h > 2$

- By construction, our rectangle has $80 \leq x \leq 160$ cells too much.
- Place the right piece (blue) to reduce field by 3.
- Place the left piece (red) to reduce field by 4.
- Use up to 3 blue pieces, and then as many red pieces as necessary.
- One can (easily) show that $80 \leq x \leq 160$ reduction is always possible.



L: Lyrical Leisure

Problem author: Lucas Schwebler

Problem

Construct a string of length n with longest palindromic substring of length k.

Solution

- Repeat character `"a"` k times as palindromic substring.
- Fill suffix with `bcdbcdbcd`...
- Python golf: `("a"*k+"bcd"*n)[:n]`

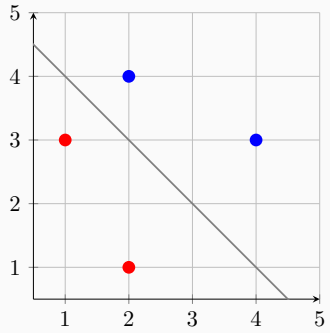
M: Mirror Magic

Problem author: Erik Sünderhauf

- **Problem:** Given two sets of points A and B. Determine whether there is a symmetry line separating A and B, i.e., mirroring A along that line results in B

- Compute the Center of Mass (COM) for both sets: $c_A = \left(\sum_{p \in A} p\right) / |A|$
- If there is a symmetry line, it
  - goes through $(c_A + c_B)/2$, and
  - is perpendicular to $c_A - c_B$

$\rightarrow$ Check that
  - $c_A \neq c_B$,
  - mirroring A results in B, and
  - all points in A lie strictly one side of the mirror line



Possible pitfalls:

- Rounding after mirroring without checking for integer coordinates
- Sorting for comparing the mirrored A with B
- Missing one of the checks above
- `float` instead of (`long`) `double`

Random facts

Jury work

- 770 secret test cases (≈ 59 per problem)
- 108 jury solutions
- The minimum number of lines the jury needed to solve all problems is

$$1 + 131 + 11 + 3 + 6 + 26 + 1 + 12 + 3 + 11 + 17 + 93 + 51 = 366$$

  On average 28.2 lines per problem
- The minimum number of characters the jury needed to solve all problems is

$$8 + 3257 + 321 + 132 + 255 + 659 + 33 + 210 + 126 + 303 + 420 + 1976 + 1430$$

  On average 702 characters per problem