Winter Contest 2022 Presentation of Solutions

January 29, 2022

Winter Contest 2022 Jury

- **Felicia Lucke**
  CPUlm
- **Nathan Maier**
  CPUlm
- **Jannik Olbrich**
  CPUlm
- **Gregor Schwarz**
  Technical University of Munich
- **Marcel Wienöbst**
  University of Lübeck
- **Paul Wild**
  Friedrich–Alexander University
  Erlangen–Nürnberg
- **Michael Zündorf**
  Karlsruhe Institute of Technology

Big thanks to our test solvers

- **Gregor Matl**
  Technical University of Munich
- **Michael Ruderer**
  CPUlm

K: Kettle Kitten

Problem Author: Jannik Olbrich, Felicia Lucke

Problem

Given a volume v and the heights and radii of many cylinders, find a smallest cylinder with volume at least v.

K: Kettle Kitten

Problem Author: Jannik Olbrich, Felicia Lucke

Problem

Given a volume v and the heights and radii of many cylinders, find a smallest cylinder with volume at least v.

Solution

- The volume V of a cylinder with height h and radius r is $V = \pi h r^2$.
- For each i calculate the volume V_i of the i-th cylinder and check whether $V_i \geq v$.
- Minimize over the volumes which are large enough.

L: Longbottom Leap

Problem Author: Jannik Olbrich

Problem

Given a binary string of length n, find the smallest integer $i \geq 1$ such that $32 \cdot 2^{i-1} \geq n$.

L: Longbottom Leap

Problem Author: Jannik Olbrich

Problem

Given a binary string of length n, find the smallest integer $i \geq 1$ such that $32 \cdot 2^{i-1} \geq n$.

Solution

Start with $i = 1$ and increment i until $32 \cdot 2^{i-1} \geq n$.

Print i times "`long`".

E: Enchanted Exam

Problem Author: Paul Wild

Problem

Find a hidden integer x $(1 \leq x \leq 100)$ using at most 50 guesses. For each guess y, you will receive one of the following replies:

- `equal`, if $y = x$;
- `factor`, if y divides x;
- `multiple`, if x divides y;
- `other`, otherwise.

E: Enchanted Exam

Problem Author: Paul Wild

Problem

Find a hidden integer x ($1 \leq x \leq 100$) using at most 50 guesses. For each guess y, you will receive one of the following replies:

- `equal`, if $y = x$;
- `factor`, if y divides x;
- `multiple`, if x divides y;
- `other`, otherwise.

Solution

- Start by guessing 2. There are four cases, depending on the answer:
  - `equal`: The hidden number is 2. Terminate.
  - `multiple`: The hidden number is 1. Guess it, then terminate.
  - `factor`: The hidden number is even. Try all 49 candidates.
  - `other`: The hidden number is odd (but not 1). Try all 49 candidates.
- Many other solutions are possible, e.g. by using prime factorization.
- Challenge: what is the least number of guesses needed in the worst case?

G: Going for Gold

Problem Author: Paul Wild

Problem

Given the rankings of n contestants in the first two events of a three-part competition, find an outcome for the third event such that contestant 1 wins. More formally:

Given are two permutations $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. Find a permutation $c_1, \ldots, c_n$ such that $a_1 b_1 c_1$ is minimal among all the $a_k b_k c_k$ $(1 \leq k \leq n)$.

G: Going for Gold

Problem Author: Paul Wild

Problem

Given the rankings of n contestants in the first two events of a three-part competition, find an outcome for the third event such that contestant 1 wins. More formally:

Given are two permutations $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. Find a permutation $c_1, \ldots, c_n$ such that $a_1 b_1 c_1$ is minimal among all the $a_k b_k c_k$ $(1 \leq k \leq n)$.

Solution

- It is always optimal if contestant 1 wins the third event, that is, if $c_1 = 1$.
- The remaining contestants should be placed in reverse order of current rank:
  - The one with the minimal $a_k b_k$ should place last ($c_k = n$).
  - ...
  - The one with the maximal $a_k b_k$ should place second ($c_k = 2$).
- If this is a valid solution, output it. Otherwise, output `impossible`.
- Time complexity: $\mathcal{O}(n \log n)$.

F: Forming Friendships

Problem Author: Marcel Wienöbst

Problem

Given a graph G, count the number of edges inserted by the following procedure: While there is a path $a - b - c$ of length two, add edge $a - c$.

F: Forming Friendships

Problem Author: Marcel Wienöbst

Problem

Given a graph G, count the number of edges inserted by the following procedure: While there is a path $a - b - c$ of length two, add edge $a - c$.

Solution

- Key insight: Each connected component of G will be transformed into a clique.
- Hence, for each connected component C, count the number of missing edges

$$\frac{1}{2} \cdot \sum_{v \in C} (|C| - \mathrm{degree}(v) - 1)$$

  and sum them all up.
- Complexity is $\mathcal{O}(|V| + |E|)$.
- Important: use 64-bit integers!

C: Cellar Chase

Problem Author: Felicia Lucke, Jannik Olbrich

Problem

Given a two-terminal-series-parallel (TTSP) graph G, find the size of a maximum cut that separates the graph into exactly two components such that two specified vertices s and t are in different components of the graph.

C: Cellar Chase

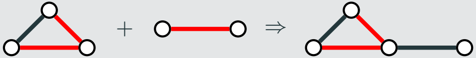
Problem Author: Felicia Lucke, Jannik Olbrich

Solution

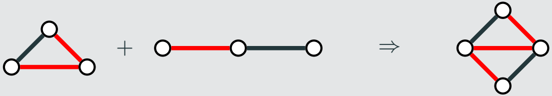
- For a graph G denote by $\text{cut}(G)$ the maximum size of a cut as defined above.
- Use the recursive structure of the graph:
  - If G is "()", $\text{cut}(G) = 1$.

  - If G is $A + B$, where A and B are both TTSP, then $\text{cut}(G) = \max(\text{cut}(A), \text{cut}(B))$.

  - If G is $A * B$, where A and B are both TTSP, then $\text{cut}(G) = \text{cut}(A) + \text{cut}(B)$.

- Calculate the size of the cut recursively.

I: Inconspicuous Identity

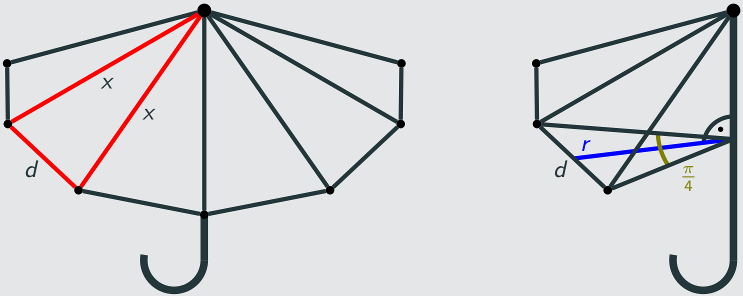
Problem Author: Gregor Schwarz

Problem

Given a square meters of fabric, compute the maximum area that can be kept dry by an umbrella which has 8 metal sticks of length x meters attached to its top.

Solution

- Check whether the amount of fabric suffices to open the umbrella all the way (i.e. metals sticks are perpendicular to the handle).
- If not, use binary search or trigonometry to compute the maximum value for d so that the fabric suffices for the umbrella.
- Given d, compute the maximum area using trigonometry.



J: Joint Jinx

Problem Author: Paul Wild

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

J: Joint Jinx

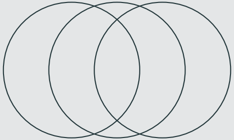
Problem Author: Paul Wild

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

Solution

- A solution exists if and only if $0 \leq k \leq n(n-1)$.
- The following construction works for all cases:



6 intersections

J: Joint Jinx

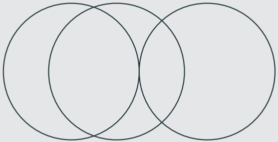
Problem Author: Paul Wild

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

Solution

- A solution exists if and only if $0 \leq k \leq n(n-1)$.
- The following construction works for all cases:



5 intersections

J: Joint Jinx

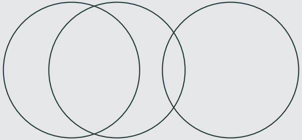
Problem Author: Paul Wild

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

Solution

- A solution exists if and only if $0 \leq k \leq n(n-1)$.
- The following construction works for all cases:



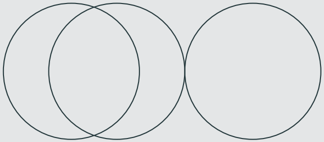
4 intersections

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

Solution

- A solution exists if and only if $0 \leq k \leq n(n-1)$.
- The following construction works for all cases:



3 intersections

Problem Author: Paul Wild

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

Solution

- A solution exists if and only if $0 \leq k \leq n(n-1)$.
- The following construction works for all cases:

2 intersections

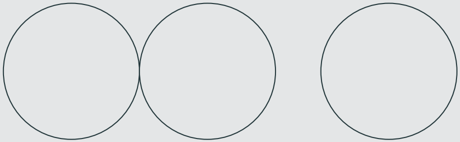
Problem Author: Paul Wild

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

Solution

- A solution exists if and only if $0 \leq k \leq n(n-1)$.
- The following construction works for all cases:



1 intersections

J: Joint Jinx

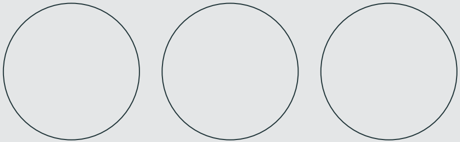
Problem Author: Paul Wild

Problem

Given integers n and k, draw n circles in the plane so that there are exactly k intersection points.

Solution

- A solution exists if and only if $0 \leq k \leq n(n-1)$.
- The following construction works for all cases:



0 intersections

B: Basic Brewing

Problem Author: Michael Zündorf

Problem

Given pairs (a, b), multiply them with a value in $[0, 1]$, sum them up such that the result is as big as possible and the ratio between a and b is x.

B: Basic Brewing

Problem Author: Michael Zündorf

Problem

Given pairs (a, b), multiply them with a value in $[0, 1]$, sum them up such that the result is as big as possible and the ratio between a and b is x.

Solution

We can partition the input into two sets:

- Those pairs with $\frac{a}{b} \geq x$
- Those with $\frac{a}{b} < x$

Observe that an optimal solution always contains all entries of one of the sets.

- Take all entries from set A and add entries from set B one by one.
- Getting as much as possible $\iff$ approach ratio x as slow as possible.
- Thus, first take entries with ratio close to x.

The total runtime is in $\mathcal{O}(n \log(n))$ to sort entries by their ratio.

A: Alohomora and Colloportus

Problem Author: Michael Zündorf

Problem

Given a Graph G, change the edges of a single vertex such that the resulting graph is a simple cycle.

A: Alohomora and Colloportus

Problem Author: Michael Zündorf

Problem

Given a Graph G, change the edges of a single vertex such that the resulting graph is a simple cycle. Alternatively, check whether G without a single vertex is a path.

A: Alohomora and Colloportus

Problem Author: Michael Zündorf

Problem

Given a Graph G, change the edges of a single vertex such that the resulting graph is a simple cycle. Alternatively, check whether G without a single vertex is a path.

Solution

We only need to check a constant number of candidate vertices:

1. One vertex with degree greater 3.
2. All vertices with degree 3 which are adjacent to all other vertices with degree 3.
3. One vertex with degree 0.
4. One vertex with degree 1.
5. One vertex.

The check if G without a vertex is a path can be done in $\mathcal{O}(n)$ and thus, the solution is in $\mathcal{O}(n)$.

M: Magic Marbles

Problem Author: Michael Zündorf

Problem

Given a string where runs of consecutive equal characters are removed if the run has length larger then k, simulate q inserts of characters into this string.

M: Magic Marbles

Problem Author: Michael Zündorf

Problem

Given a string where runs of consecutive equal characters are removed if the run has length larger then k, simulate q inserts of characters into this string.

Solution

- You just need to simulate this efficiently.
- Either use a *treap* and keep track of run lengths.
- Or a *binary search tree* which contains runs.
- In both cases your data structure needs to efficiently do this:
  - Insert a character at a position.
  - Find the length of a run at a position.

Total runtime $\mathcal{O}(q \log(n))$

D: Document Dimensions

Problem Author: Michael Zündorf

Problem

Given a text with n words separated by spaces with total length W, replace some spaces with newlines such that the total height plus width of the text is minimized.

D: Document Dimensions

Problem Author: Michael Zündorf

Problem

Given a text with n words separated by spaces with total length W, replace some spaces with newlines such that the total height plus width of the text is minimized.

Solution

- For a given width w we can find the minimal height greedily by only adding newlines when needed.
- The next position where a newline is needed can be found in $\mathcal{O}(1)$ with a prefix sum over the lengths of the words.
- Therefore, the minimal height can be found in $\mathcal{O}(\frac{W}{w})$.

D: Document Dimensions

Problem Author: Michael Zündorf

Problem

Given a text with n words separated by spaces with total length W, replace some spaces with newlines such that the total height plus width of the text is minimized.

Solution

- For a given width w we can find the minimal height greedily by only adding newlines when needed.
- The next position where a newline is needed can be found in $\mathcal{O}(1)$ with a prefix sum over the lengths of the words.
- Therefore, the minimal height can be found in $\mathcal{O}(\frac{W}{w})$.
- Calculating this for every width is in $\mathcal{O}(W \log(W))$.

H: Hidden Horcrux

Problem Author: Gregor Schwarz

Problem

Determine the number of water carriers that Harry needs to travel d days through the desert. Each person can carry c units of water but needs to drink 1 water unit a day.

H: Hidden Horcrux

Problem Author: Gregor Schwarz

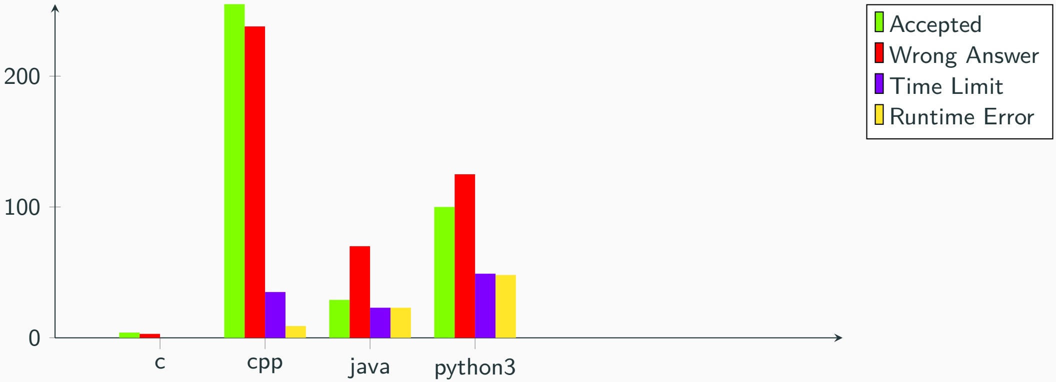
Problem

Determine the number of water carriers that Harry needs to travel d days through the desert. Each person can carry c units of water but needs to drink 1 water unit a day.

Solution

- Distribute water among carriers so that Harry reaches day $d - c$ and still has full water capacity.
- From day $d - c$ onward, Harry travels alone.
- $c - 2 \geq d - c$ must hold so that the last water carrier can return home.
- Simulate how far Harry can get with n water carriers. Binary search the minimum value for n.
- Alternative: Start at day $d - c$ with only one water carrier. Move the timeline backwards and add additional water carriers when necessary.

Language stats



Random facts

Jury work

- 260 commits

Random facts

Jury work

- 260 commits
- 522 secret test cases ($\approx$ 40 per problem)

Random facts

Jury work

- 260 commits
- 522 secret test cases (≈ 40 per problem)
- 103 jury solutions

Random facts

Jury work

- 260 commits
- 522 secret test cases (≈ 40 per problem)
- 103 jury solutions
- The minimum number of lines the jury needed to solve all problems is

  $$32 + 36 + 13 + 31 + 7 + 29 + 9 + 11 + 9 + 16 + 7 + 4 + 69 = 273$$

  On average 21 lines per problem