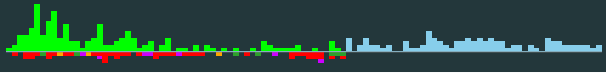
NWERC 2022 Test Session

Solutions presentation

November 26, 2022

C: Cup Covering

Problem Author: The NWERC 2022 jury

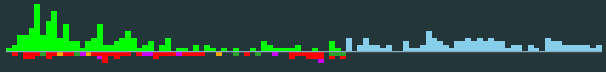


Problem

Given the area a of a stroopwafel, calculate the diameter of a cup that the stroopwafel can perfectly cover.

C: Cup Covering

Problem Author: The NWERC 2022 jury



Problem

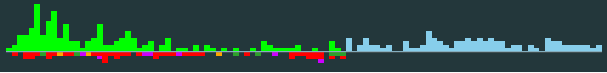
Given the area a of a stroopwafel, calculate the diameter of a cup that the stroopwafel can perfectly cover.

Solution

- The area of a circle is $a = \pi r^2$, and the diameter is twice the radius, $d = 2r$.

C: Cup Covering

Problem Author: The NWERC 2022 jury



Problem

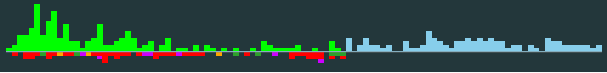
Given the area a of a stroopwafel, calculate the diameter of a cup that the stroopwafel can perfectly cover.

Solution

- The area of a circle is $a = \pi r^2$, and the diameter is twice the radius, $d = 2r$.
- Rewriting this gives $d = 2\sqrt{\frac{a}{\pi}}$.

C: Cup Covering

Problem Author: The NWERC 2022 jury



Problem

Given the area a of a stroopwafel, calculate the diameter of a cup that the stroopwafel can perfectly cover.

Solution

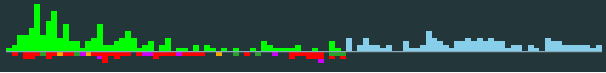
- The area of a circle is $a = \pi r^2$, and the diameter is twice the radius, $d = 2r$.
- Rewriting this gives $d = 2\sqrt{\frac{a}{\pi}}$.

Pitfall

`floats` do not cover a precision of 10^{-9}, so you need to use `doubles`.

C: Cup Covering

Problem Author: The NWERC 2022 jury



Problem

Given the area a of a stroopwafel, calculate the diameter of a cup that the stroopwafel can perfectly cover.

Solution

- The area of a circle is $a = \pi r^2$, and the diameter is twice the radius, $d = 2r$.
- Rewriting this gives $d = 2\sqrt{\frac{a}{\pi}}$.

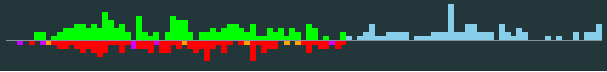
Pitfall

`floats` do not cover a precision of 10^{-9}, so you need to use `doubles`.

Statistics: 345 submissions, 153 accepted, 101 unknown

A: An Interactive Problem

Problem Author: Maarten Sijm

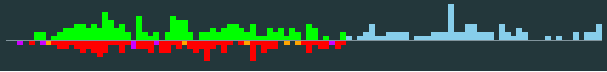


Problem

Find the highest value in a square of unknown size with upside-down lots.

A: An Interactive Problem

Problem Author: Maarten Sijm



Problem

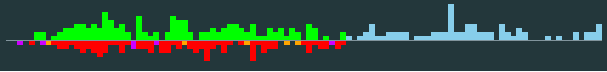
Find the highest value in a square of unknown size with upside-down lots.

Solution

- Start by querying the value of $(x, 1)$ for every x starting from 1, until you get "`ArrayIndexOutOfBoundsException`" as response.
  - Now you know the size of the square n: the last x for which you got an integer response.

A: An Interactive Problem

Problem Author: Maarten Sijm



Problem

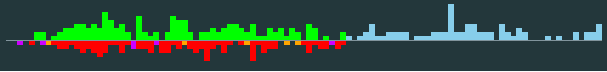
Find the highest value in a square of unknown size with upside-down lots.

Solution

- Start by querying the value of $(x, 1)$ for every x starting from 1, until you get "`ArrayIndexOutOfBoundsException`" as response.
  - Now you know the size of the square n: the last x for which you got an integer response.
- Query all positions (x, y) for $1 \leq x \leq n$ and $2 \leq y \leq n$.

A: An Interactive Problem

Problem Author: Maarten Sijm



Problem

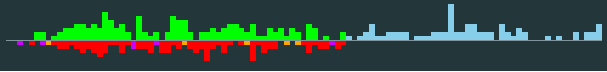
Find the highest value in a square of unknown size with upside-down lots.

Solution

- Start by querying the value of $(x, 1)$ for every x starting from 1, until you get "`ArrayIndexOutOfBoundsException`" as response.
  - Now you know the size of the square n: the last x for which you got an integer response.
- Query all positions (x, y) for $1 \leq x \leq n$ and $2 \leq y \leq n$.
- Print the highest value found.

A: An Interactive Problem

Problem Author: Maarten Sijm



Problem

Find the highest value in a square of unknown size with upside-down lots.

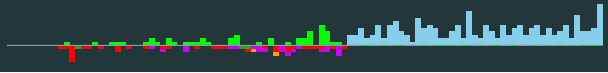
Solution

- Start by querying the value of $(x, 1)$ for every x starting from 1, until you get "`ArrayIndexOutOfBoundsException`" as response.
  - Now you know the size of the square n: the last x for which you got an integer response.
- Query all positions (x, y) for $1 \leq x \leq n$ and $2 \leq y \leq n$.
- Print the highest value found.

Statistics: 339 submissions, 138 accepted, 78 unknown

B: Brothers in Arms

Problem Author: Markus Himmel



Problem

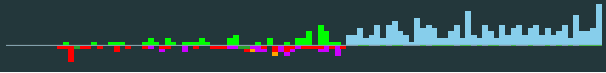
Given $n \leq 90\,000$ medieval cities and $s \leq 300$ possible symbols, compute the friendship degree between $q \leq 10^5$ pairs of cities.

Two cities are

- directly friendly if they share a symbol at the top or share a symbol at the bottom on their coats of arms, or
- indirectly friendly through a chain of direct friendships, with *friendship degree* defined as the minimum length of that chain.

B: Brothers in Arms

Problem Author: Markus Himmel



Problem

Given $n \leq 90\,000$ medieval cities and $s \leq 300$ possible symbols, compute the friendship degree between $q \leq 10^5$ pairs of cities.

Two cities are

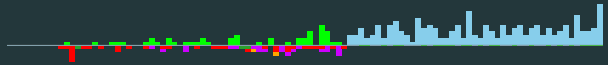
- directly friendly if they share a symbol at the top or share a symbol at the bottom on their coats of arms, or
- indirectly friendly through a chain of direct friendships, with *friendship degree* defined as the minimum length of that chain.

Observations

- Treating symbols for the top and bottom differently gives at most 600 symbols in total.

B: Brothers in Arms

Problem Author: Markus Himmel



Problem

Given $n \leq 90\,000$ medieval cities and $s \leq 300$ possible symbols, compute the friendship degree between $q \leq 10^5$ pairs of cities.

Two cities are

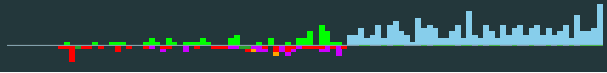
- directly friendly if they share a symbol at the top or share a symbol at the bottom on their coats of arms, or
- indirectly friendly through a chain of direct friendships, with *friendship degree* defined as the minimum length of that chain.

Observations

- Treating symbols for the top and bottom differently gives at most 600 symbols in total.
  - Give each symbol a unique identifier. E.g., by adding 300 to the identifiers of the bottom symbols.

B: Brothers in Arms

Problem Author: Markus Himmel



Problem

Given $n \leq 90\,000$ medieval cities and $s \leq 300$ possible symbols, compute the friendship degree between $q \leq 10^5$ pairs of cities.

Two cities are

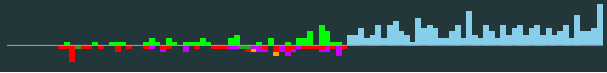
- directly friendly if they share a symbol at the top or share a symbol at the bottom on their coats of arms, or
- indirectly friendly through a chain of direct friendships, with *friendship degree* defined as the minimum length of that chain.

Observations

- Treating symbols for the top and bottom differently gives at most 600 symbols in total.
  - Give each symbol a unique identifier. E.g., by adding 300 to the identifiers of the bottom symbols.
- A graph with cities as vertices and edges as direct friendships is too large to compute all queries.

B: Brothers in Arms

Problem Author: Markus Himmel



Problem

Given $n \leq 90\,000$ medieval cities and $s \leq 300$ possible symbols, compute the friendship degree between $q \leq 10^5$ pairs of cities.

Two cities are

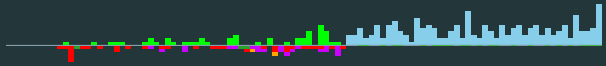
- directly friendly if they share a symbol at the top or share a symbol at the bottom on their coats of arms, or
- indirectly friendly through a chain of direct friendships, with *friendship degree* defined as the minimum length of that chain.

Observations

- Treating symbols for the top and bottom differently gives at most 600 symbols in total.
  - Give each symbol a unique identifier. E.g., by adding 300 to the identifiers of the bottom symbols.
- A graph with cities as vertices and edges as direct friendships is too large to compute all queries.
  - However, there are only 600 symbols...

B: Brothers in Arms

Problem Author: Markus Himmel



Problem

Given $n \leq 90\,000$ medieval cities and $s \leq 300$ possible symbols, compute the friendship degree between $q \leq 10^5$ pairs of cities.

B: Brothers in Arms

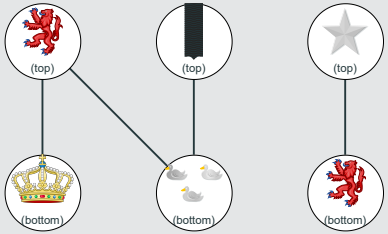
Problem Author: Markus Himmel

Problem

Given $n \leq 90\,000$ medieval cities and $s \leq 300$ possible symbols, compute the friendship degree between $q \leq 10^5$ pairs of cities.

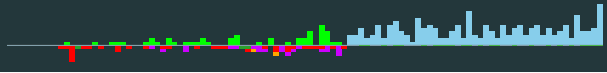
Solution

- Create a graph where the vertices represent symbols and the edges represent cities: for every city, add a bidirectional edge between its top and bottom symbol.



B: Brothers in Arms

Problem Author: Markus Himmel

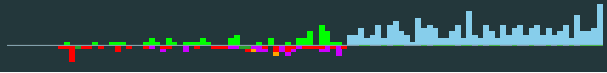


Solution

- Create a graph where the vertices represent symbols and the edges represent cities: for every city, add a bidirectional edge between its top and bottom symbol.
- Store as distance matrix with entries:
  - $d(a, a) = 0$ for every symbol a,
  - $d(a, b) = d(b, a) = 1$ for every city with symbols a and b,
  - $d(x, y) = \infty$ else.

B: Brothers in Arms

Problem Author: Markus Himmel

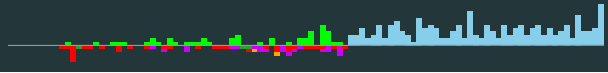


Solution

- Create a graph where the vertices represent symbols and the edges represent cities: for every city, add a bidirectional edge between its top and bottom symbol.
- Store as distance matrix with entries:
  - $d(a, a) = 0$ for every symbol a,
  - $d(a, b) = d(b, a) = 1$ for every city with symbols a and b,
  - $d(x, y) = \infty$ else.
- Find the distances between all pairs of symbols using the Floyd–Warshall algorithm ($\mathcal{O}(s^3)$).

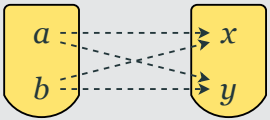
B: Brothers in Arms

Problem Author: Markus Himmel



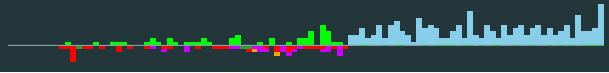
Solution

- Create a graph where the vertices represent symbols and the edges represent cities: for every city, add a bidirectional edge between its top and bottom symbol.
- Store as distance matrix with entries:
  - $d(a, a) = 0$ for every symbol a,
  - $d(a, b) = d(b, a) = 1$ for every city with symbols a and b,
  - $d(x, y) = \infty$ else.
- Find the distances between all pairs of symbols using the Floyd–Warshall algorithm ($\mathcal{O}(s^3)$).
- For every query of a city with symbols a and b and a city with symbols x and y:
  - Compute $\min(d(a, x), d(a, y), d(b, x), d(b, y))$ to get their friendship degree ($\mathcal{O}(1)$ per query).
  - If the minimum distance is ∞, print -1.



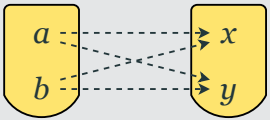
B: Brothers in Arms

Problem Author: Markus Himmel



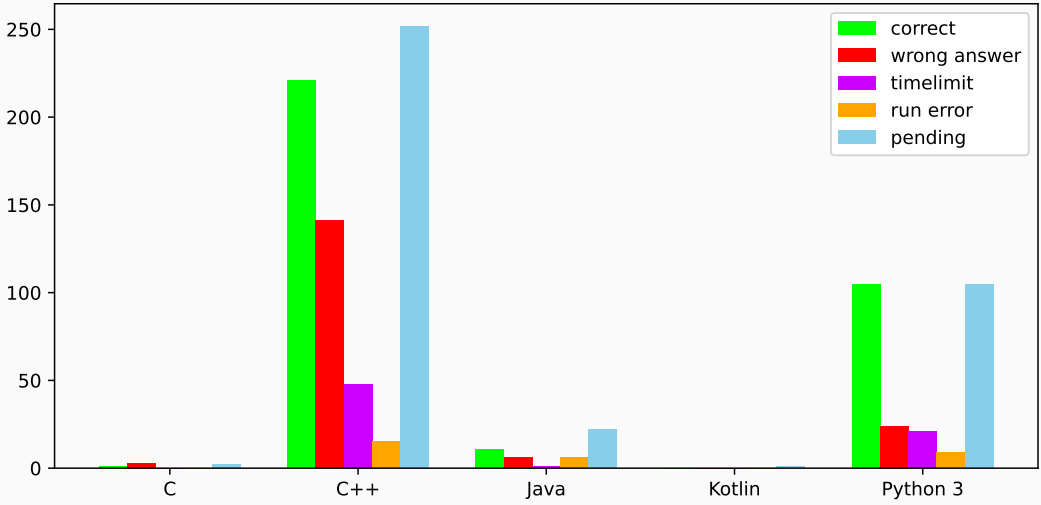
Solution

- Create a graph where the vertices represent symbols and the edges represent cities: for every city, add a bidirectional edge between its top and bottom symbol.
- Store as distance matrix with entries:
  - $d(a, a) = 0$ for every symbol a,
  - $d(a, b) = d(b, a) = 1$ for every city with symbols a and b,
  - $d(x, y) = \infty$ else.
- Find the distances between all pairs of symbols using the Floyd–Warshall algorithm ($\mathcal{O}(s^3)$).
- For every query of a city with symbols a and b and a city with symbols x and y:
  - Compute $\min(d(a, x), d(a, y), d(b, x), d(b, y))$ to get their friendship degree ($\mathcal{O}(1)$ per query).
  - If the minimum distance is ∞, print -1.



Statistics: 331 submissions, 47 accepted, 203 unknown

Language stats



Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).
- The judging is typically case-insensitive (no guarantees though).

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).
- The judging is typically case-insensitive (no guarantees though).
- How to flush standard output: please check the documentation of your programming language before tomorrow.

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).
- The judging is typically case-insensitive (no guarantees though).
- How to flush standard output: please check the documentation of your programming language before tomorrow.
- You *will not* get a time penalty if your submission has a compilation error.

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).
- The judging is typically case-insensitive (no guarantees though).
- How to flush standard output: please check the documentation of your programming language before tomorrow.
- You *will not* get a time penalty if your submission has a compilation error.

Tomorrow:

- you will get three copies of the problem set;

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).
- The judging is typically case-insensitive (no guarantees though).
- How to flush standard output: please check the documentation of your programming language before tomorrow.
- You *will not* get a time penalty if your submission has a compilation error.

Tomorrow:

- you will get three copies of the problem set;
- you are *NOT* allowed to have your bags or any electronic equipment on you (except for medical reasons);

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).
- The judging is typically case-insensitive (no guarantees though).
- How to flush standard output: please check the documentation of your programming language before tomorrow.
- You *will not* get a time penalty if your submission has a compilation error.

Tomorrow:

- you will get three copies of the problem set;
- you are *NOT* allowed to have your bags or any electronic equipment on you (except for medical reasons);
- you *MUST* wear your shirt and badge visibly;

Clarifications

Clarifications

- Compiler flags can be found at 2022.nwerc.eu/systems and in `~/.bashrc`
- The memory limit of your submission is 2 GiB (also see 2022.nwerc.eu/systems).
- The judging is typically case-insensitive (no guarantees though).
- How to flush standard output: please check the documentation of your programming language before tomorrow.
- You *will not* get a time penalty if your submission has a compilation error.

Tomorrow:

- you will get three copies of the problem set;
- you are *NOT* allowed to have your bags or any electronic equipment on you (except for medical reasons);
- you *MUST* wear your shirt and badge visibly;
- after the contest, you must take everything with you.

Systems update

Systems update

- Printing from CodeBlocks is probably broken.

Systems update

Systems update

- Printing from CodeBlocks is probably broken.
- The print command is `printfile <file>`