German Collegiate Programming Contest 2018
June 16th

Problems

A Attack on Alpha-Zet
B Battle Royale
C Coolest Ski Route
D Down the Pyramid
E Expired License
F Fighting Monsters
G GPS
H Hyper Illuminati
I It's Time for a Montage
J Jigsaw Puzzle
K Kitchen Cable Chaos
L Logic Puzzle
M Mountaineers

This page is intentionally left (almost) blank.

Problem A: Attack on Alpha-Zet

Space pirate Captain Krys has recently acquired a map of the artificial and highly secure planet Alpha-Zet which he has been planning to raid for ages. It turns out the whole planet is built on a 2D plane with modules that serve as one room each. There is exactly one module at every pair of integer coordinates and modules are exactly 1×1 units big. Every module is bidirectionally connected to at least one adjacent module. Also, for any two modules there exists exactly one path between them. All in all the modules create a rectangular maze without any loops.
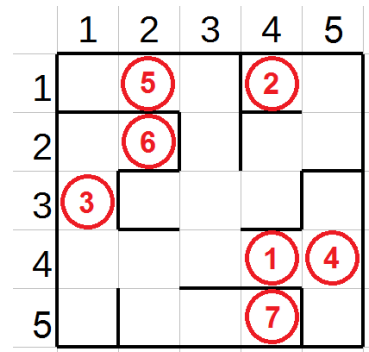

Figure A.1: Illustration of Sample Input 2

On the map Captain Krys has marked several modules he wants to visit in exactly the marked order. What he intends to do there is none of your business, but he promises you a fortune if you determine the number of modules he has to walk through along the route (since there are no loops he will always take the direct route from one marked module to the next). The first marked module indicates where he starts his journey, the last where he wants to finish.

Input

The input consists of:

- one line with two integers h and w ($2 \le h, w \le 1\,000$) describing the height and the width of the maze.
- $h + 1$ lines follow, describing the maze in ASCII, each line containing $2 \cdot w + 1$ characters. The description always follows these rules:
  - In every row, columns with odd index (starting at index 1) contain either vertical walls or spaces and columns with even index contain either horizontal walls or spaces.
  - The first row describes the northern wall of the maze (which always consists only of horizontal walls). Every subsequent row describes a row of modules.
  - A module is located at every even column index. Its western and eastern walls are located at the directly neighboring odd column indices respectively, its northern wall is located at the same column index but one row above and its southern wall can be found at its own position. If a wall is missing, the corresponding position contains a space instead.
- After the description of the maze, an integer m ($2 \le m \le 10^4$) is given.
- Each of the following m lines describes a marked module with two integer coordinates x and y ($1 \le x \le h$; $1 \le y \le w$). The first pair of coordinates is the start point of the journey, the last pair the end point. Modules may appear multiple times but never twice or more in a row. $(1, 1)$ is the top left module and (h, w) is the bottom right module.

It is guaranteed that the maze itself is enclosed. Furthermore it is guaranteed that exactly one path exists between any two modules.

Output

Output one integer, the number of modules Captain Krys has to travel through if he follows the route in the exact order given in the input.

Sample Input 1

```
2 6
 _ _ _ _ _ _
|  _ _ _ _ _|
|_ _ _ _ _ _|
5
1 5
1 1
1 6
1 1
1 5
```

Sample Output 1

```
18
```

Sample Input 2

```
5 5
 _ _ _ _ _
|_ _   |_   |
|   _| |  _|
| |_    _| |
|     _ _   |
|_|_ _ _|_|
7
4 4
1 4
3 1
4 5
1 2
2 2
5 4
```

Sample Output 2

```
43
```

Problem B: Battle Royale

Battle Royale games are the current trend in video games and *Gamers Concealed Punching Circles* (GCPC) is the most popular game of them all. The game takes place in an area that, for the sake of simplicity, can be thought of as a two-dimensional plane. Movement and positioning are a substantial part of the gameplay, but getting to a desired location can be dangerous. You are confident in your ability to handle the other players, however, while you are walking to your destination, there are two hazards posed by the game itself:

- The game zone is bounded by a blue circle. Outside of this circle, there is a deadly force field that would instantly take you out of the game.
- Inside the game zone, there is a red circle where you are exposed to artillery strikes. This circle is also too risky to enter.

You want to move from one spot on the map to another, but the direct path to your destination is blocked by the red circle, so you need to find a way around it. Can you find the shortest path that avoids all hazards by never leaving the blue or entering the red circle? Touching the boundaries of the circles is fine, as long as you do not cross them.

Input

The input consists of:

- one line with two integers x_c, y_c specifying your current location;
- one line with two integers x_d, y_d specifying your destination;
- one line with three integers x_b, y_b, r_b specifying the center and radius of the blue circle;
- one line with three integers x_r, y_r, r_r specifying the center and radius of the red circle.

All coordinates have an absolute value of at most $1\,000$, and $1 \le r_b, r_r \le 1\,000$. The red circle is strictly inside the blue circle. Your current location and destination are strictly inside the blue circle and strictly outside of the red circle, and the direct path between them is blocked by the red circle.

Output

Output the length of the shortest path that does not leave the blue or enter the red circle. The output must be accurate up to a relative or absolute error (whichever is lower) of 10^{-7}.

Sample Input 1

```
0 0
10 0
0 0 1000
5 0 2
```

Sample Output 1

```
10.8112187742
```

This page is intentionally left (almost) blank.

Problem C: Coolest Ski Route

John loves winter. Every skiing season he goes heli-skiing with his friends. To do so, they rent a helicopter that flies them directly to any mountain in the Alps. From there they follow the picturesque slopes through the untouched snow.

Of course they want to ski on only the best snow, in the best weather they can get. For this they use a combined condition measure and for any given day, they rate all the available slopes.

Can you help them find the most awesome route?

Input

The input consists of:

- one line with two integers n ($2 \le n \le 1000$) and m ($1 \le m \le 5000$), where n is the number of (1-indexed) connecting points between slopes and m is the number of slopes.
- m lines, each with three integers s, t, c ($1 \le s, t \le n$, $1 \le c \le 100$) representing a slope from point s to point t with condition measure c.

Points without incoming slopes are mountain tops with beautiful scenery, points without outgoing slopes are valleys. The helicopter can land on every connecting point, so the friends can start and end their tour at any point they want. All slopes go downhill, so regardless of where they start, they cannot reach the same point again after taking any of the slopes.

Output

Output a single number n that is the maximum sum of condition measures along a path that the friends could take.
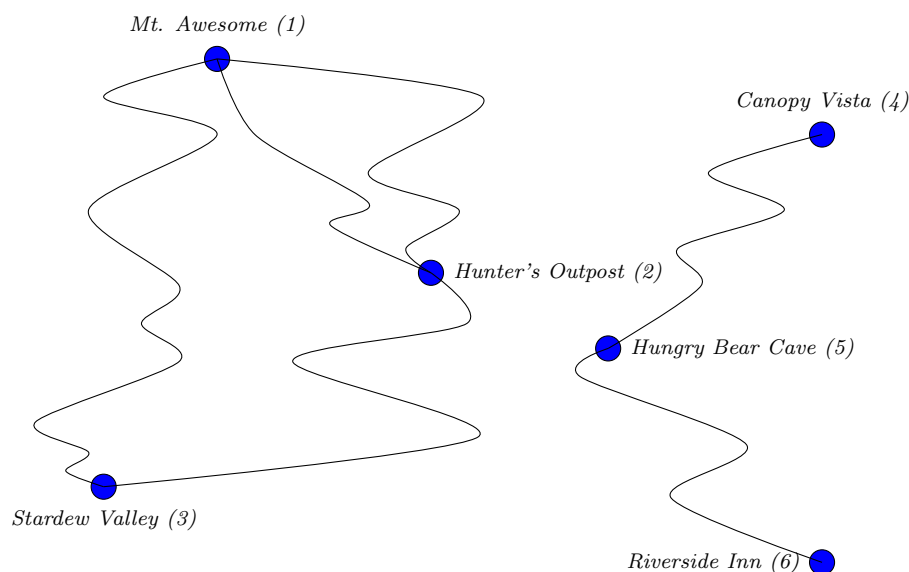
Sample visualization

Figure C.1: Map of the second sample case

Sample Input 1

```
5 5
1 2 15
2 3 12
1 4 17
4 2 11
5 4 9
```

Sample Output 1

```
40
```

Sample Input 2

```
6 6
1 2 2
4 5 2
2 3 3
1 3 2
5 6 2
1 2 4
```

Sample Output 2

```
7
```

Problem D: Down the Pyramid

Do you like number pyramids? Given a number sequence that represents the base, you are usually supposed to build the rest of the "pyramid" bottom-up: For each pair of adjacent numbers, you would compute their sum and write it down above them. For example, given the base sequence $[1, 2, 3]$, the sequence directly above it would be $[3, 5]$, and the top of the pyramid would be $[8]$:

```
        ┌───────┐
        │   8   │
    ┌───┴───┬───┴───┐
    │   3   │   5   │
┌───┴───┬───┴───┬───┴───┐
│   1   │   2   │   3   │
└───────┴───────┴───────┘
```
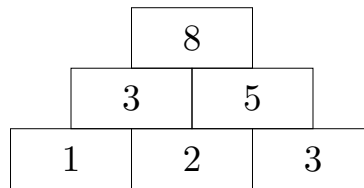
However, I am not interested in completing the pyramid – instead, I would much rather go underground. Thus, for a sequence of n non-negative integers, I will write down a sequence of $n + 1$ non-negative integers below it such that each number in the original sequence is the sum of the two numbers I put below it. However, there may be several possible sequences or perhaps even none at all satisfying this condition. So, could you please tell me how many sequences there are for me to choose from?

Input

The input consists of:

- one line with the integer n ($1 \le n \le 10^6$), the length of the base sequence.
- one line with n integers $a_1, \ldots, a_n$ ($0 \le a_i \le 10^8$ for each i), forming the base sequence.

Output

Output a single integer, the number of non-negative integer sequences that would have the input sequence as the next level in a number pyramid.

Sample Input 1	Sample Output 1
6 12 5 7 7 8 4	2

Sample Input 2	Sample Output 2
3 10 1000 100	0

This page is intentionally left (almost) blank.

Problem E: Expired License

Paul is an extremely gifted computer scientist who just completed his master's degree at a prestigious German university. Now he would like to culminate his academic career in a PhD. The problem is that there are so many great universities out there that it is hard for him to pick the best. Because some application deadlines are coming up soon, Paul's only way to procrastinate his decision is by simply applying to all of them.

Most applications require Paul to attach a portrait photo. However, it seems like there does not exist an international standard for the aspect ratio of these kinds of photos. While most European universities ask Paul to send a photograph with aspect ratio 4.5 by 6, some Asian countries discard the applications immediately if the photo does not have an aspect ratio of 7.14 by 11.22, precisely.

As Paul has never been interested in photo editing, he never had a reason to spend a lot of money on proper software. He downloaded a free trial version some months ago, but that version has already expired and now only works with some funny restrictions. The cropping tool, for example, no longer accepts arbitrary numbers for setting the aspect ratio, but only primes. This makes Paul wonder whether the desired aspect ratios can even be properly expressed by two prime numbers. Of course, in case this is possible, he would also like to know the primes he has to enter.

Input

The input consists of:

- one line with an integer n ($1 \leq n \leq 10^5$), the number of applications Paul has to file;
- n lines, each with two real numbers a and b ($0 < a, b < 100$), where $a \times b$ is the desired aspect ratio of one application.

All real numbers are given with at most 5 decimal places after the decimal point.

Output

For each application, if it is possible to represent the desired aspect ratio by two prime numbers p and q, output one line with p and q. Otherwise, output `impossible`. If multiple solutions exist, output the one minimizing $p + q$.

Sample Input 1

```
3
4.5 6
7.14 11.22
0.00002 0.00007
```

Sample Output 1

```
impossible
7 11
2 7
```

This page is intentionally left (almost) blank.

Problem F: Fighting Monsters

Emma just discovered a new card game called *Gwint: A wizard's game*. There are two types of cards: monster cards and spell cards. Monster cards are used to score points, while spell cards typically interact with the monsters in some way.

On each monster card there is an integer value, the *power* of the monster. Monsters can fight each other, and during these fights the power acts as both the strength and the health of the monster. The monsters take turns hitting each other until one of them dies. Whenever a monster A hits a monster B, this causes B to lose an amount of power equal to the power of A. Conversely, if B hits A, A loses power equal to the power of B (see the example below). This continues until one of the two monsters has a power of zero or less, at which point this monster is considered dead.
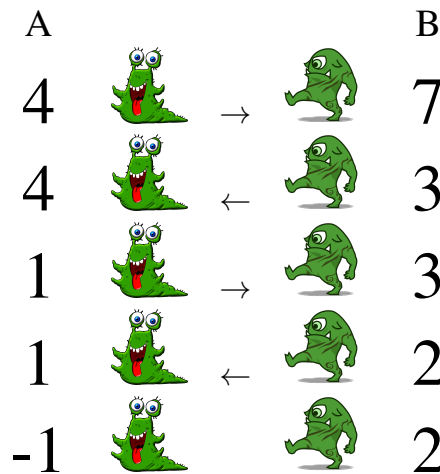
Images by OpenClipart-Vectors on Pixabay and from PhantomOpenEmoji.

Figure F.1: A fight between monsters A and B, starting with powers of 4 and 7, respectively. A hits first. B wins with a remaining power of 2.

One of Emma's most beloved cards in the game is a spell called *Fight!* which states:

> Pick two monsters. They fight each other to the death. If the surviving monster has a power of exactly 1 left, return this card to your hand.

Of course, Emma would like to play as efficiently as possible by picking two monsters such that *Fight!* is returned to her hand. However, there are often a lot of monsters on the board, which makes it very time consuming to figure out whether this can be done or not. Can you help her find two monsters she can pick so that she gets the card back?

Input

The input consists of:
- one line with an integer n ($2 \leq n \leq 10^5$), the number of monsters;
- one line with n integers $m_1, \ldots, m_n$ ($1 \leq m_i \leq 10^6$), giving the power of each monster.

Output

If there is no pair of monsters that Emma can pick, output `impossible`. Otherwise, output two distinct integers i, j ($1 \leq i, j \leq n$), where i is the index of the monster that starts the fight and j is the index of the other monster. If multiple solutions exist, any of them will be accepted.

Sample Input 1

```
4
1 12 67 8
```

Sample Output 1

```
impossible
```

Sample Input 2

```
5
1 1 12 67 8
```

Sample Output 2

```
2 1
```

Sample Input 3

```
6
1 5 6 7 90 8
```

Sample Output 3

```
2 6
```

Problem G: GPS

You, like so many other people, have probably used the Global Positioning Systems (GPS), but have wondered how it actually works. GPS positioning is based on a set of satellites orbiting Earth. All satellites have a synchronized internal clock[1]. They continuously transmit the value of their internal clock to the surface of Earth via radio waves. A GPS receiver simultaneously[2] collects measurements from some satellites which are in sight. The time measurements $t_1, \ldots, t_m$ collected at time t are typically different, as each satellite has a different distance to the receiver. Since we know the exact orbits of all satellites, we know exactly where each satellite was when it sent its signal. Based on these positions and the distance, we can compute our position. To compute a unique position, assuming that we are on the surface of the Earth, we need at least 3 measurements from satellites.

Your task, however, is not to actually compute your own position, but to determine which signals you received from which satellites. To be precise: you are given your geographical coordinates on Earth and a description of all orbits of satellites and should now determine for every satellite

- whether a signal sent from the satellite at a certain time will reach you (which is the case if it does not pass through the Earth).
- if the signal does reach you, the time it took for the signal to travel from the satellite to your position.

Remember that radio waves travel in straight lines and at the speed of light, which we assume is 299 792 458 meters per second. We assume Earth to be a perfect sphere with a radius of 6 371km, and that all GPS satellites are orbiting the Earth with a fixed speed on perfect circles whose centers are identical to the center of Earth.

Input

The input consists of:

- one line with an integer s ($1 \le s \le 10^4$), the number of satellites;
- one line with two real numbers ℓ_o, ℓ_a, the longitude and latitude of your position;
- s lines, each with four real numbers ϕ, ψ, r, x describing one of the satellites.

For the longitude $\ell_o \in [-180, 180]$, positive values represent east and negative values west.
For the latitude $\ell_a \in [-90, 90]$, positive values represent north and negative values south.

The format for the satellite descriptions is as follows:

- $\phi \in [-180, 180]$ – the reference longitude at which the orbit intersects with the Earth's equator (each orbit intersects the equator (at least) twice). Positive longitude represents east, negative longitude represents west.
- $\psi \in [0, 360]$ – the angle between orbit and the Earth's eastbound equator when passing through the equator at longitude ϕ. Values between 0 and 180 indicate northbound movement while values between 180 and 360 indicate southbound movement. If the orbit is coplanar to the equator, the angle ψ will be 0 or 180.
- $r \in [7\,000, 10^6]$ – the radius of the satellite's orbit in kilometers.
- $x \in [0, 1)$ – the fraction of the orbit the satellite had covered at the time when its signal was sent, starting from its intersection with the equator at ϕ.

---

[1]In practice the clocks are not completely synchronous – because of general relativity. In this problem we assume that these effects do not exist.
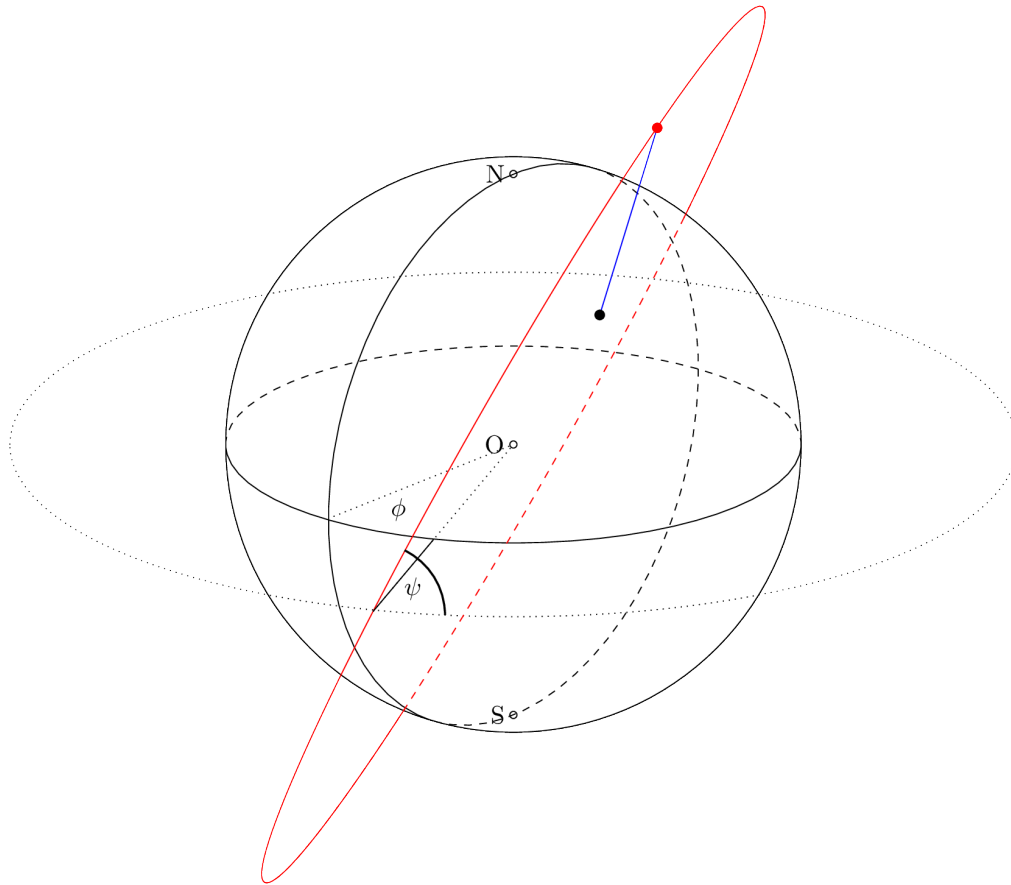
[2]Another simplification.

Figure G.1: Schematic display of satellite coordinates. The satellite's orbit is shown in red, its actual position is the red dot. On Earth, the equator and the prime meridian are marked by lines. The dotted line is coplanar to the equator and has the same radius as the satellite's orbit. The angles ϕ and ψ are described in the input section. The black dot is your position on Earth and you should compute the time necessary for the signal to travel the blue line.

Note: All real numbers in the input are given with at most 3 decimal places after the decimal point. It is guaranteed that each satellite has a distance of at least 1 meter from the tangent plane at your position (this means that you can safely use floating point arithmetic in your solution).

Output

For every satellite, output `no signal` if a signal sent from that satellite does not reach you. If the satellite's signal does reach you, output the time that the signal needs to travel from the satellite to your position, in seconds. The time must be accurate up to a relative or absolute error (whichever is lower) of 10^{-7}.

Sample Input 1	**Sample Output 1**
3	0.00773245
45.0 45.0	no signal
0.0 45.0 7500.0 0.125	0.0690288
0.0 60.0 7000.0 0.3	
-50.0 5.0 25000.0 0.3	

Problem H: Hyper Illuminati

Once again the time dawns to demonstrate the sheer power of the Illuminati. To do so, it was decided to build an n-dimensional hyper-step pyramid using n-dimensional *blocks*:

- All the steps of the pyramid are n-dimensional hyper-cuboids.
- Every step has a height of exactly 1 *block* in the n-th dimension.
- The pyramid has s steps and the base step is s *blocks* long in every other of the $n - 1$ dimensions.
- Every subsequent higher step is 1 *block* shorter in each of the $n - 1$ dimensions than the step below it.
- The top step is exactly 1 *block*.

To prove their might even further the Illuminati leaders have decided to add two more requirements:

- n must be at least 3.
- The number of *blocks* used to build the pyramid must be a meaningful number.
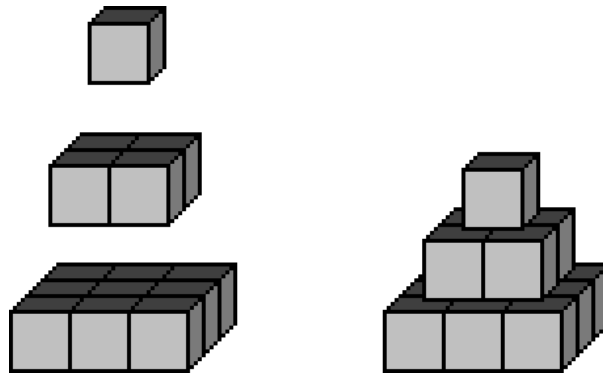
Figure H.1: A 3-dimensional hyper pyramid with 3 steps consisting of 14 *blocks* in total.

Input

The input consists of:

- one line with a single integer m $(1 \le m \le 10^{16})$. This integer is the meaningful number the leaders have chosen.

Output

If a hyper-step pyramid matching all the requirements exists, output a single line with two integers n and s, the dimension of the pyramid and its number of steps. If none exists, output `impossible`. If multiple solutions exist, any will be accepted.

Sample Input 1

```
14
```

Sample Output 1

```
3 3
```

Sample Input 2

```
9
```

Sample Output 2

```
4 2
```

Sample Input 3

```
24
```

Sample Output 3

```
impossible
```

Sample Input 4

```
9134731356568978
```

Sample Output 4

```
5 2147
```

Problem I: It's Time for a Montage

The heroes of your favorite action TV show are preparing for the final confrontation with the villains. Fundamentally, there are two rivals who will fight each other: a very important main hero who wants to save the universe and an equally important main villain who wants to destroy it. However, through countless recursive spin-offs, they may have slightly less important sidekicks (a hero and a villain who are rivals themselves), who in turn may also have their own (even less important) sidekicks, and so on. Note that there is an equal number of heroes and villains, and each rival pair has at most one sidekick pair.

Initially, every character will fight their rival, with the winner being determined by who has the higher *Power Level*. If a hero and their corresponding villain have the same Power Level, their battle will be determined by their sidekicks' battle, as the winning sidekick can help as a sort of tiebreaker. (If rivals of equal Power Level do not have sidekicks, the hero character will win with the help of random passersby.) However, whenever a battle is won by either side, there is nothing the sidekicks can do about it – this is because the people behind the show believe some fans might get upset if a character were to get defeated by a bunch of less important characters, so they would lose regardless of the Power Levels.

After the battles between rivals (and possible tiebreakers) are done, the most important character remaining will defeat the rest of the opposing side and determine the fate of the universe. Fortunately, the heroes can ensure victory through hard, rigorous training. For each day they spend training, the Power Level of each hero increases by 1, while the villains' Power Levels remain constant.

But you already knew all this. The question plaguing your mind is how long the training is going to take.

Input

The input consists of:

- one line with an integer n ($1 \leq n \leq 1\,000$), giving the number of rival pairs.
- one line with n integers $h_1, \ldots, h_n$ ($1 \leq h_i \leq 1\,000$ for each i), the i-th value giving the Power Level of the i-th most important hero.
- one line with n integers $v_1, \ldots, v_n$ ($1 \leq v_i \leq 1\,000$ for each i), the i-th value giving the Power Level of the i-th most important villain.

Output

Output a single integer, the minimum number of days the heroes need to spend training in order for their side to win.

Sample Input 1

```
4
5 3 1 1
8 6 9 1
```

Sample Output 1

```
4
```

Sample Input 2

```
1
2
1
```

Sample Output 2

```
0
```

Sample Input 3

```
2
4  2
7  5
```

Sample Output 3

```
3
```

Problem J: Jigsaw Puzzle

You found a box with old games when cleaning up your attic, and among them was also a jigsaw puzzle. Unfortunately, the packaging was damaged, so a couple of puzzle pieces are scattered around the bottom of the box, and you suspect that some of the pieces may have been lost elsewhere. In fact, given the orderliness of your attic, some of the pieces in the box may even come from some entirely different puzzle! So now you have a pile of puzzle pieces lying in front of you and you are trying to assemble them into a solved puzzle.
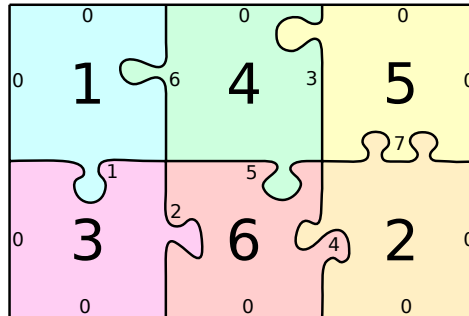
Figure J.1: Illustration of the first sample.

More formally:

- There are n square-shaped pieces, numbered from 1 to n, which need to be arranged side by side to form a single rectangle. All n pieces have to be used.
- The edges of the pieces are either straight or irregular. Straight edges must be placed on the boundary of the assembled rectangle and irregular edges must be placed on the inside.
- The irregular edges are all shaped differently, so that each edge shape occurs exactly two times, on two different puzzle pieces. Two pieces can only be placed next to each other if the shapes of the corresponding edges match.
- You may rotate the pieces, but you may not flip them over.

Input

The input consists of:

- one line with one integer n ($1 \leq n \leq 3 \cdot 10^5$), the number of pieces;
- n lines, each with four integers, the i-th line gives the connections (edge shapes) of the i-th piece in counter-clockwise order.

A connection of type 0 stands for a straight edge. The other connection types are numbered with consecutive positive integers starting from 1 and each of them occurs exactly two times, on two different lines.

Output

If the pieces cannot be assembled as described above, output `impossible`. Otherwise, output the solved puzzle in the following format:

- one line with two integers h, w ($h, w \geq 1$, $h \cdot w = n$), the height and width of the grid;
- h lines, each with w integers, the numbers of the pieces.

Any rotation of the correct solution will by accepted.

Sample Input 1

```
6
0 0 1 6
0 7 4 0
0 0 2 1
5 3 0 6
3 7 0 0
4 5 2 0
```

Sample Output 1

```
2 3
1 4 5
3 6 2
```

Sample Input 2

```
4
0 0 1 2
0 0 2 3
0 0 3 4
0 0 1 4
```

Sample Output 2

```
impossible
```

Problem K: Kitchen Cable Chaos

You started your new project: installing a home automation system. You already bought all the components and in your local electronic shop you found a promotion set with a bunch of cables of different lengths. Now you want to connect your controller with your smart sandwich maker that is several meters away, but you are lacking a cable that is long enough.

To solve this problem you have to connect some of your cables to form a long one. You measure the lengths of every cable you own. Exactly 5 centimeters of isolation are stripped on both ends of every cable. To connect two cables, you overlap and twist the stripped ends. It is enough for the cables to touch each other with an overlap of 0. You cannot have an overlap of more than 5 centimeters, but the connection quality increases with longer overlaps. On both ends – the controller and the sandwich maker – you also have 5 centimeters of stripped end, to which you have to connect your newly created cable in the same way. The connection quality of your link is determined by the smallest overlap used and your goal is to maximize this value.

The problem would be really easy, but your perfectionist roommate hates unnecessary use of cables. Therefore, the cable has to form a straight line, without any loops or detours. And cutting the cables is no option, obviously.
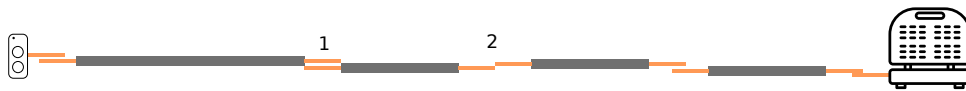
Figure K.1: Four cables of different lengths connect the controller with the sandwich maker, with different overlaps. Connection 1 has the maximal overlap, connection 2 the minimal overlap and all other connections are in between. The quality of this setup is 0.

Considering all possible arrangements of cables, find the one with the best quality.

Input

The input consists of:

- one line with two integers n, g ($1 \le n \le 60$, $11 \le g \le 1\,000$), the number of cables and the distance to be covered in centimeters, measured between the casings of the controller and the sandwich maker;
- n lines, each with an integer d ($11 \le d \le 1\,000$), giving the lengths of the cables (including the stripped ends).

Each of the n cables can be used at most once.

Output

Output one number, the best achievable quality. The quality must be accurate up to a relative or absolute error (whichever is lower) of 10^{-7}. If no arrangement fits your needs, output `impossible`.

Sample Input 1

```
3 70
20
35
50
```

Sample Output 1

```
3.3333333
```

```
3 150                         impossible
20
35
50
```

Problem L: Logic Puzzle

While browsing a kiosk at a recent trip, you bought a magazine filled with various kinds of logic puzzles. After a while of solving, however, you start to get a bit bored of the puzzles. Still wanting to complete all the puzzles in the magazine, you start wondering about ways to solve some of them algorithmically.

The puzzle you are currently trying to solve is called *Mosaic*, and it is quite similar to the classic *Minesweeper* video game:

Figure L.1: Illustration of the first sample

You are given a two-dimensional grid of cells, initially all white, and you have to color some of the cells in black. You are also given a grid of clue numbers, which extends beyond the borders of the puzzle grid by one cell in each direction. The number in a cell indicates (exactly) how many cells in the 3×3 block centered at this cell need to be colored in black. You may not color any cells outside of the original grid.

Input

The input consists of:

- one line with two integers h, w ($1 \le h, w \le 100$), the height and width of the puzzle;
- $h + 2$ lines, each with $w + 2$ integers $c_1, \ldots, c_{w+2}$ ($0 \le c_i \le 9$), the clue numbers.

Output

If the given clue numbers are inconsistent, output `impossible`. Otherwise, output h lines with w characters each, the solution to the puzzle. Use `X` for black cells and `.` for white cells. If there are multiple solutions, any of them will be accepted.

Sample Input 1

```
2 3
1 1 2 1 1
1 2 3 2 1
1 2 3 2 1
0 1 1 1 0
```

Sample Output 1

```
X.X
.X.
```

Sample Input 2

```
1 2
0 0 1 1
0 1 1 1
0 1 1 1
```

Sample Output 2

```
impossible
```

Problem M: Mountaineers

The Chilean Andes have become increasingly popular as a destination for backpacking and hiking. Many parts of the Andes are quite remote and thus dangerous. Because of this, the Ministry of Tourism wants to help travelers plan their trips. In particular, the travelers need to know how high they will have to climb during their journey, as this information will help them decide which equipment they need to bring. The Ministry has tasked you to provide the aspiring mountaineers with this data.

You are given a topographic map of a part of the Andes, represented as a two-dimensional grid of height values, as well as the list of origins and destinations. Mountaineers can move from each grid cell to any of the four adjacent cells. For each mountaineer find the minimal height that they must be able to reach in order to complete their journey.

Input

The input consists of:

- one line with three integers m, n and q ($1 \leq m, n \leq 500, 1 \leq q \leq 10^5$), where m is the number of rows, n is the number of columns, and q is the number of mountaineers;
- m lines, each with n integers $h_1, \ldots, h_n$ ($1 \leq h_i \leq 10^6$), the height values in the map;
- q lines, each with four integers x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 \leq m, 1 \leq y_1, y_2 \leq n$), describing a mountaineer who wants to trek from (x_1, y_1) to (x_2, y_2).

The top left cell of the grid has coordinates $(1, 1)$ and the bottom right cell has coordinates (m, n).

Output

Output q integers, the minimal height for each mountaineer, in the same order as in the input.

Sample Input 1

```
3 5 3
1 3 2 1 3
2 4 5 4 4
2 1 3 2 2
1 1 3 2
2 4 2 2
1 4 3 4
```

Sample Output 1

```
2
4
3
```

This page is intentionally left (almost) blank.

GCPC 2018
Presentation of solutions

Statistics
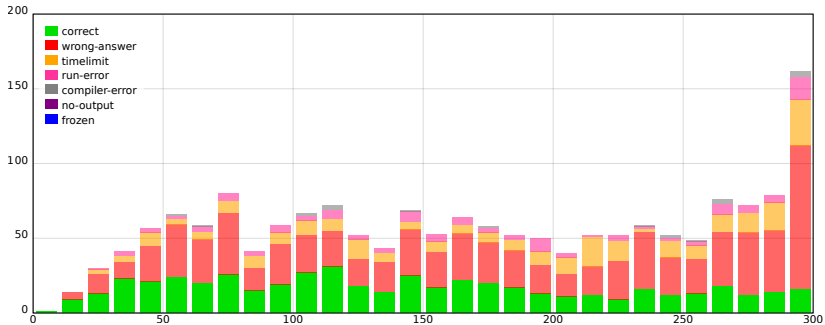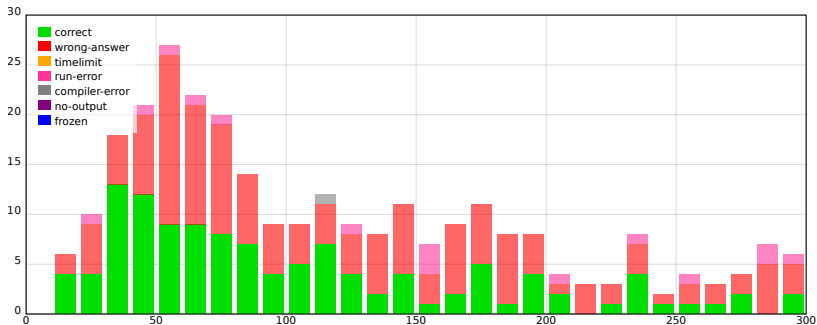
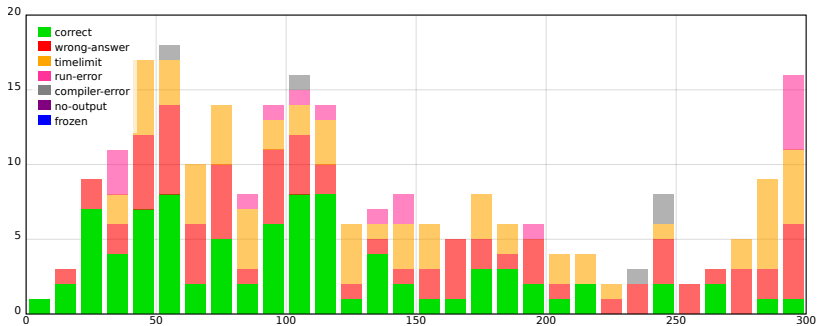# I – It's Time for a Montage

Problem

Given two integer sequences H and V of equal length, find the minimal non-negative d to add to the values in H such that *lexicographically*, $H + d \geq V$.

Solution

- If $H_1 > V_1$, the answer is 0.
- Otherwise, put $d := V_1 - H_1$.
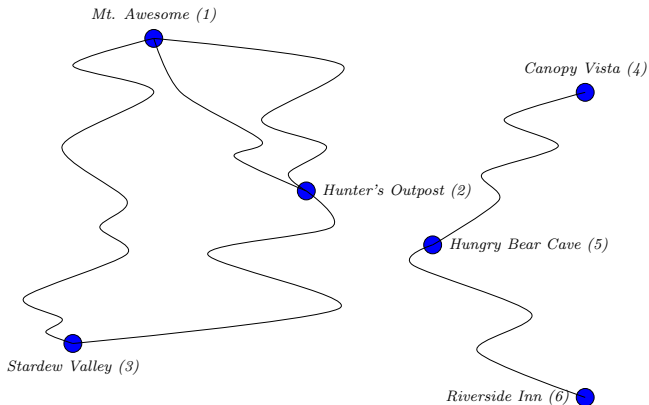- If $H + d < V$, the answer is $d + 1$, else it is d.

C – Coolest Ski Route

C – Coolest Ski Route

Problem

Given a graph of ski slopes labelled with *condition measures*, find a path with maximal sum of these measures.

C – Coolest Ski Route

Problem

Given a graph of ski slopes labelled with *condition measures*, find a path with maximal sum of these measures.

Solution

- The graph of ski slopes is a *directed acyclic graph*.
- Find the *longest path* between any two nodes.
- Multiple graph algorithms can be used for this:
    - Invert the measures, use Floyd-Warshall ($\mathcal{O}(n^3)$).
    - Add a super source, invert the measures, then use Bellman-Ford for shortest paths ($\mathcal{O}(n * m)$).
    - Find a topological ordering, then find the maximum for each node with dynamic programming ($\mathcal{O}(n + m)$).

F – Fighting Monsters
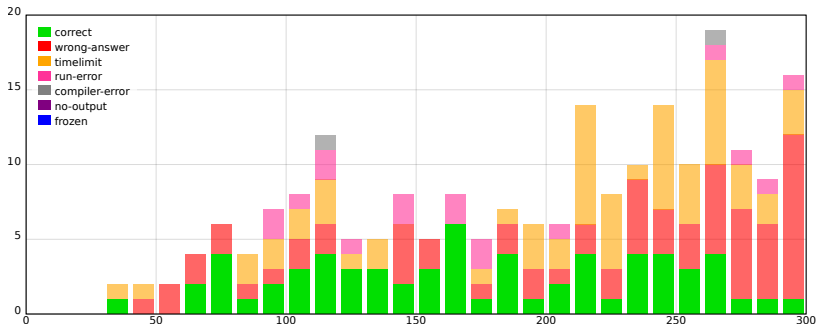
F – Fighting Monsters

Problem

Given a list of numbers, are there two numbers such that alternating subtraction of the two values results in a pair $(0, 1)$?

Observation

We calculate backwards from the final state:

$$(0, 1) \leftarrow (1, 1) \leftarrow (1, 2) \leftarrow (3, 2) \leftarrow (3, 5) \leftarrow (8, 5) \ldots$$

These are Fibonacci numbers!

Solution

- Generate Fibonacci numbers up to 10^6.
- Check if two consecutive Fibonacci numbers appear in the list.
- Be careful about monsters with power 1.
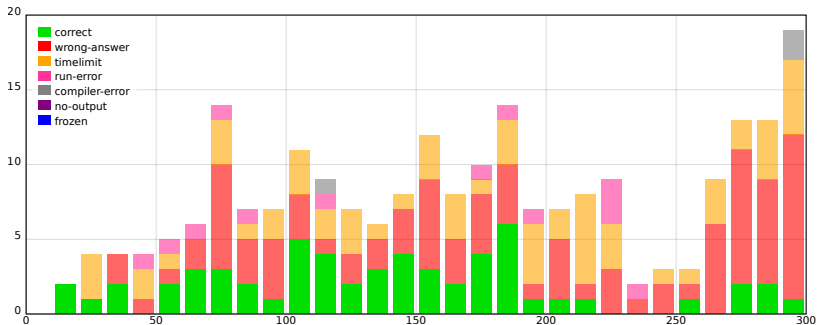
D – Down the Pyramid

Problem

Given a non-negative integer sequence A, find the number of possible non-negative integer sequences B such that $A_i = B_i + B_{i+1}$ for all pairs of adjacent numbers in B.

Insights

- B_1 uniquely determines the rest of the sequence.
- If B_1 increases, each odd-numbered element increases and each even-numbered element decreases.

Solution

- Put $B_1 := 0$ and compute the resulting sequence.
- Find the minimum odd- and even-numbered elements.
- The range of answers is $[-\min\{B_{2i+1}\}, \min\{B_{2i}\}]$.
- Time complexity: $\mathcal{O}(n)$.

Problem

Given a rectangular grid of numbers, color some of the cells in black such that the number in each cell equals the number of adjacent black cells. Cells on the boundary may not be colored.

1	1	2	1	1
1	2	3	2	1
1	2	3	2	1
0	1	1	1	0

1	1	2	1	1
1	2	3	2	1
1	2	3	2	1
0	1	1	1	0
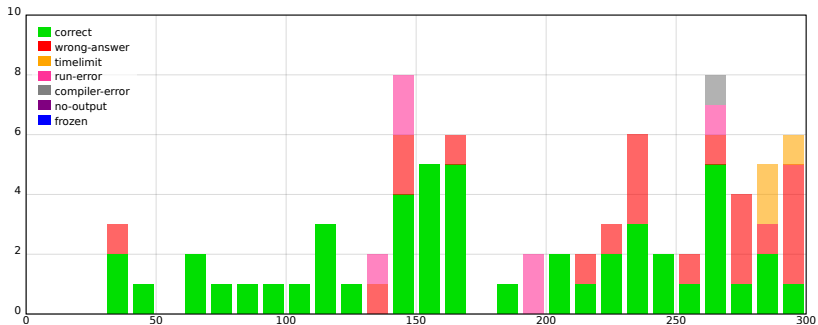
L – Logic Puzzle

Problem

Given a rectangular grid of numbers, color some of the cells in black such that the number in each cell equals the number of adjacent black cells. Cells on the boundary may not be colored.

Solution

- The solution can be reconstructed row by row.
- For every cell, check if the number to the top left is positive.
- If it is, put an X and subtract 1 from every adjacent cell.
- Do a second pass to check if any non-zero numbers remain.
- If there are no conflicts, output the solution.

E – Expired License
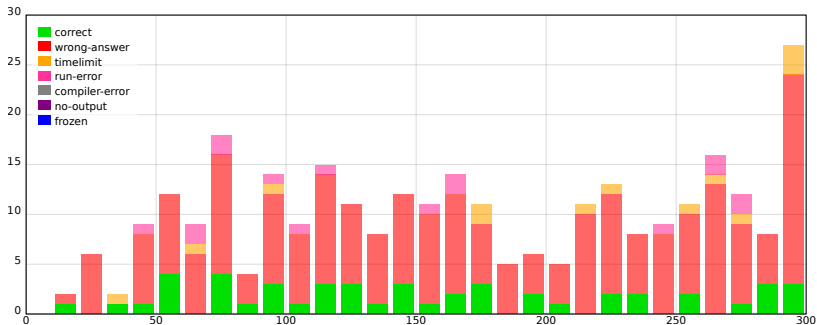
E – Expired License

Problem

Given two floating point numbers representing the aspect ratio $a \times b$ of a photo, find prime numbers p and q such that $\frac{a}{b} = \frac{p}{q}$.

Solution

- Multiply a and b by 10^5 to make both numbers integral.
- Divide both numbers by their greatest common divisor.
- Check whether the resulting two numbers are prime, e.g. using the Sieve of Eratosthenes.
- Special case: quadratic aspect ratios, i.e. 6×6.
- Take care of numerical issues when using floating point arithmetics, e.g. $0.00007 \cdot 10^5 = 6.9999999999999991118\ldots$

H – Hyper Illuminati
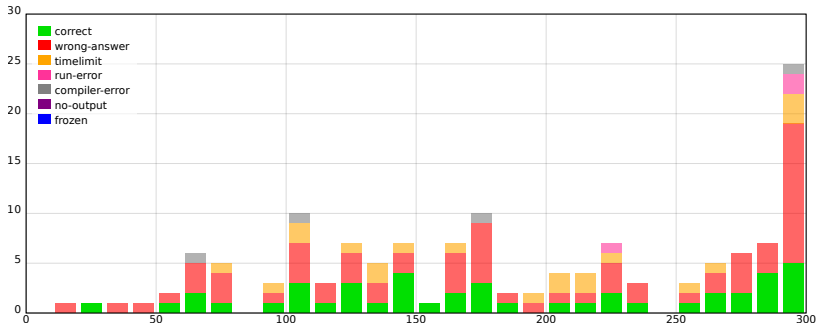
H – Hyper Illuminati

Problem

Build an *n*-dimensional step-pyramid consisting of *m* blocks.

Observation

- We need to find a dimension n and a step number s such that

$$\sum_{k=1}^{s} k^{n-1} = m.$$

- The k^{n-1} grow very fast, especially for higher dimensions.

H – Hyper Illuminati

Solution

Calculate all candidate pyramids keeping the following in mind:

- We can stop increasing s as soon as $s^{n-1} > m$.
- We can stop increasing n as soon as $2^{n-1} > m$.

In total, there are only $328\,373$ pyramids with two or more steps and at most 10^{16} blocks.

Possible pitfall: avoid the pow() function in C++ and Java.

B – Battle Royale
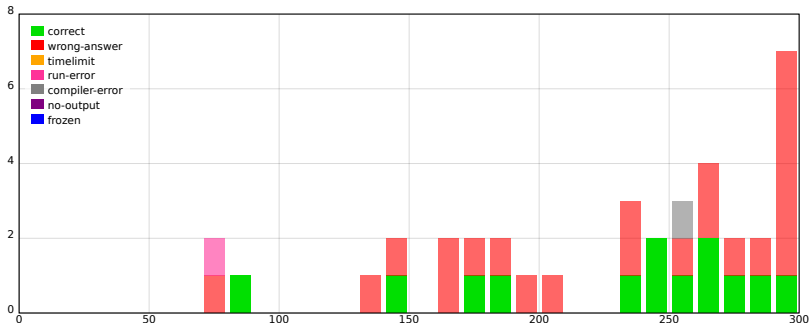
B – Battle Royale

Problem

Given a **red circle**, a **blue circle** and **two points**, find the length of the shortest path between the two points while staying inside the blue circle and outside the red circle.

Simplifications:

- Touching the circles is allowed.
- The direct path is always blocked by the red circle.
- The red circle and the two points are completely inside the blue circle.
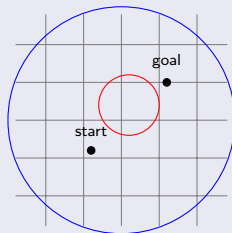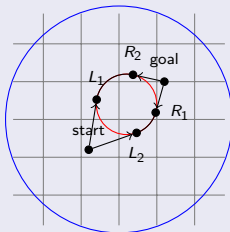
B – Battle Royale
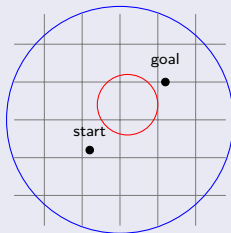
Problem

Deductions

- The blue circle is irrelevant.
- No need two check if the direct connection is possible.

B – Battle Royale

Problem

Solution

- Compute **tangents** of the red circle going through start and end.
- Compute the length of **circle segments** between the touching points $L_{1,2}$, $R_{1,2}$.
- Find the minimum length of the four paths
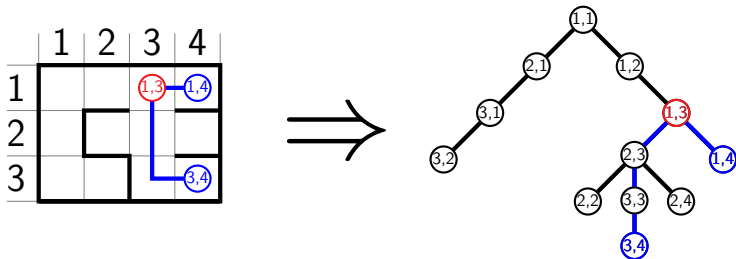 $start \rightarrow L_{1,2} \rightarrow R_{1,2} \rightarrow end$.
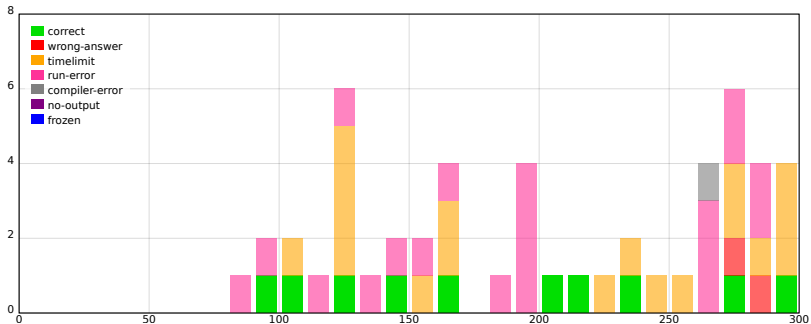
A – Attack on Alpha-Zet

Problem

Given a sequence of locations inside a maze, what is the minimal distance one has to travel to visit them all in given order?

Observation

- The number of locations is too big for multiple iterations of any search algorithm.
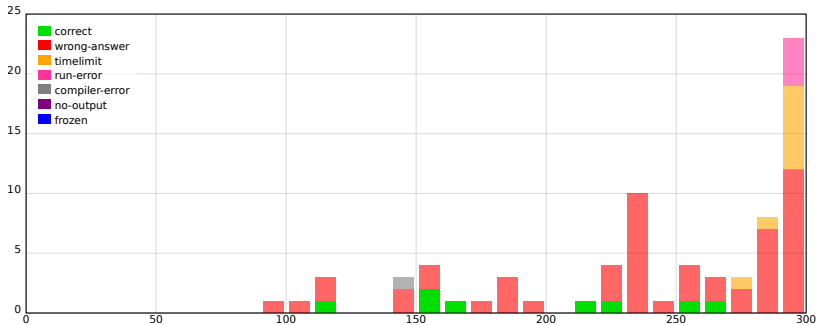- Since the maze has no loops, it can be seen as a tree.

A – Attack on Alpha-Zet

Solution

- Transform the maze into a tree e.g. with *depth first search*.
- Find *Lowest Common Ancestors* to calculate the distance between two consecutive locations.
- $\Rightarrow$ Init: $\mathcal{O}(n \log n)$, Lookup: $\mathcal{O}(1)$

K – Kitchen Cable Chaos

K – Kitchen Cable Chaos

Problem

Given n cables with lengths $d_1, \ldots, d_n$, find the set of cables that has the *largest minimal overlap* if fitted in the gap g.

Solution

- Calculate all pairs (i, j) such that there is a set of i cables with total length j.
- This can be done with a knapsack-like DP in $\mathcal{O}(n^2 \cdot g)$.
- The largest minimal overlap for a pair (i, j) is given by $\frac{j + 10 - g}{i + 1}$.
- Try all pairs to find the best solution.
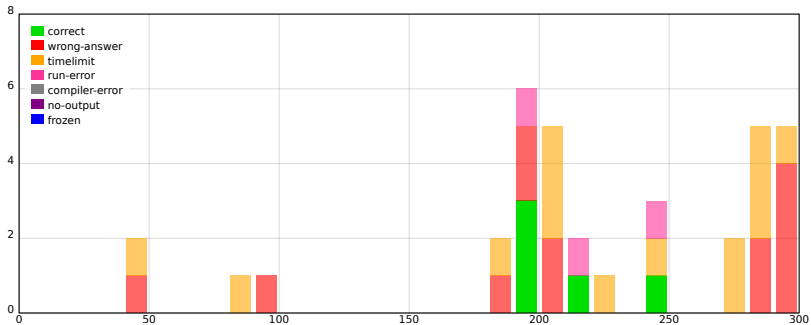
M – Mountaineers

Problem

Given are some pairs of points in a topographic map. For each pair find the least maximal height of a path connecting both ends.

Intuition

- Imagine a rising water line throughout the mountain range.
- The answer for a pair is the lowest height at which it becomes possible to swim from one end to the other.

M – Mountaineers

Solution

- Store connected components in a union-find data structure. In each component, store a list of end points.
- Merge neighboring cells by increasing height.
- While merging:
  - Always merge the smaller list into the larger list.
  - If both ends of a pair are in the two lists, the current height is the answer for that pair.
- Total time complexity: $\mathcal{O}(k \cdot \log^2 k)$ where $k = \max(m \cdot n, q)$.
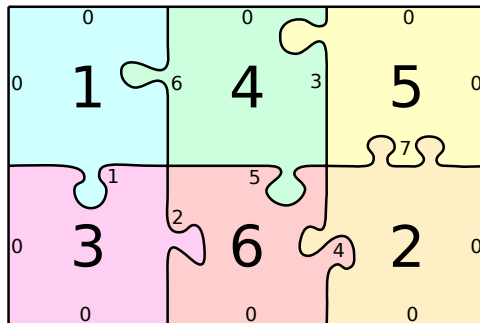
J – Jigsaw Puzzle
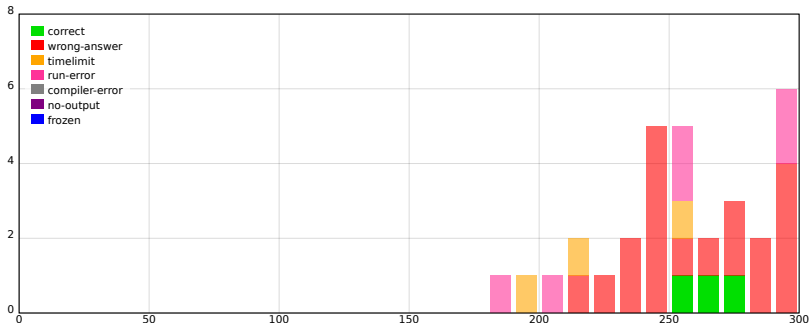
J – Jigsaw Puzzle

Problem

Assemble a jigsaw puzzle from square shaped pieces with jagged edges.

J – Jigsaw Puzzle

Problem

Assemble a jigsaw puzzle from square shaped pieces with jagged edges.

Solution 1

- Pick an arbitrary piece and place it at $(0, 0)$.
- Run a breadth-first-search from this piece, fixing coordinates and orientation of the other pieces along the way.
- Check that this is a valid solution (no gaps/overlaps/...).
- Each step can be done in time $\mathcal{O}(n)$.

Solution 2

Find a corner piece and reconstruct the solution row by row.
Be *very* careful when checking the connections.

G – GPS

G – GPS
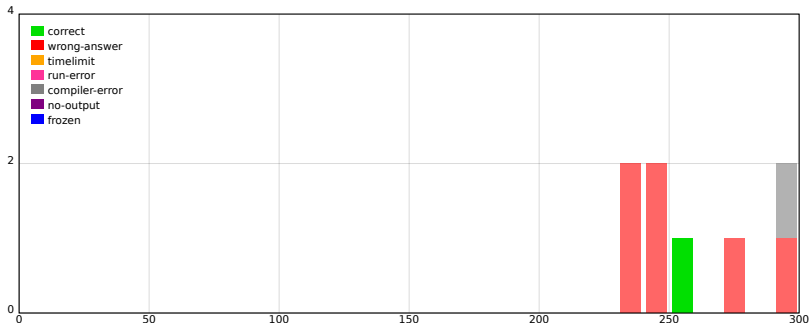
Problem

Given a point $\vec{x}$ on a sphere of radius r and a point $\vec{y}$ outside the sphere, compute whether the line $\overrightarrow{xy}$ intersects the sphere and if not, output $|\overrightarrow{xy}|$.

Solution, part II

Assume $\vec{x}$ and $\vec{y}$ are given in cartesian coordinates (x, y, z).
To check whether the line $\overrightarrow{xy}$ intersects the sphere:

- Compute closest point $\vec{p}$ on the line to $\vec{0} = (0, 0, 0)$ (vector projection, "Lot fällen"). You can use your 2D-Formula for this. It is also correct in higher dimensions.
- Check whether $|\vec{p}|$ is larger than r. If not, the line will intersect the sphere.

Alternatively, check on which side of the tangential plane $\vec{y}$ lies (normal vector of the plane is $\vec{x}$) using dot product.

Solution, part I

Points are not given in cartesian coordinates ...

1. $\vec{x}$: Given in terms of geographic coordinates (ℓ_o, ℓ_a). These are (similar to) polar coordinates – their conversion is known.

2. $\vec{y}$: Circle that intersects the XY-plane at (at least) two points with an angle ψ. One of the points is at longitude ϕ. The point is where one has covered $x\%$ of the circle from ϕ.

Compute the coordinate stepwise:

- Start with circle in equatorial plane of correct radius. Find point on $x\%$ of the circle.
- Rotate in YZ-plane by ψ. Note that the orbit intersection is still at longitude 0
- Rotate in XY-plane by ϕ.