Solutions – IDI Open 2021

March 20th 2021

The Judges

(In alphabetical order)

- Dmitry Ivankov
- Edvard Kristoffer Neset Karlsen
- Jean Niklas L'orange
- Nils Barlaug
- Sander Land
- Sondre Sortland

Want to join us? Send an email to `jeannikl@hypirion.com` or send a message to `hyPiRion` here on Discord!

Jumbo Javelin

- Sum up integers, but subtract 1 per addition
-
- $\text{sum}(xs) - (xs.length - 1)$

Solved by 21 teams
First solution after 2 minutes

Damaged Equation

- Find all equations on the form $a\ ?\ b\ =\ c\ ?\ d$ where a, b, c and d are given, and ? can be $+$, $-$, $*$ or $/$ (integer division).
- Solution: Try all combinations! There are only 16, after all.
- To sort alphanumerically, it's sufficient to sort the strings before printing them.

Solved by 20 teams
First solution after 10 minutes

Damaged Equation

- Find all equations on the form $a\ ?\ b\ =\ c\ ?\ d$ where a, b, c and d are given, and ? can be $+$, $-$, $*$ or $/$ (integer division).
- Solution: Try all combinations! There are only 16, after all.
- To sort alphanumerically, it's sufficient to sort the strings before printing them.
- Note: Dividing by 0 is *not* allowed. Making it return -1 or 0 won't hold.
- Example: $1/0 = 1/0$ is not a valid expression.

Solved by 20 teams
First solution after 10 minutes

Bootstrapping Number

- Find x in the equation $x^x = n$, given n.

Solved by 20 teams
First solution after 2 minutes

Bootstrapping Number

- Find x in the equation $x^x = n$, given n.
- First find some number u such that $u^u \geq n$.
  - $10^{10} = 10\,000\,000\,000$, so 10 is the upper limit.
  - Also possible to try $u = 1, 2, 3, \ldots$ until $u^u \geq n$

Solved by 20 teams
First solution after 2 minutes

Bootstrapping Number

- Find x in the equation $x^x = n$, given n.
- First find some number u such that $u^u \geq n$.
  - $10^{10} = 10\,000\,000\,000$, so 10 is the upper limit.
  - Also possible to try $u = 1, 2, 3, \ldots$ until $u^u \geq n$
- Then binary search between 0 and u until the absolute or relative error is small enough to be accepted
  - This happens the distance between the upper bound for x and lower bound for x is smaller than 10^{-6}
  - You can also check the distance between n and x^x, but that has to be a relative check. You'll get into an infinite loop if you check for $10^{-6} < |n - x^x|$
  - Or you realise that 10 000 iterations will be more than sufficient and ignore the pesky details altogether

Solved by 20 teams

First solution after 2 minutes

Eternal Embers

Author: Jean Niklas L'orange

"The Worst Best Firefighter", from buttersafe.com. Cropped.

- Given a graph G, find the shortest way to extinguish all fires in the Downstairs, then head back to the start.

Solved by 12 teams
First solution after 27 minutes

Eternal Embers

Author: Jean Niklas L'orange

"The Worst Best Firefighter", from buttersafe.com. Cropped.

- Given a graph G, find the shortest way to extinguish all fires in the Downstairs, then head back to the start.
- Use Floyd-Warshall to find the shortest path from all fires to one another. Now we have the Travelling Salesman problem!

Solved by 12 teams
First solution after 27 minutes

Eternal Embers

Author: Jean Niklas L'orange

"The Worst Best Firefighter", from buttersafe.com. Cropped.

- Given a graph G, find the shortest way to extinguish all fires in the Downstairs, then head back to the start.
- Use Floyd-Warshall to find the shortest path from all fires to one another. Now we have the Travelling Salesman problem!
- With fires $N = 12$, brute force ($\mathcal{O}(N!)$) would be too slow. Use well-knowns DP tricks to speed it up to $\mathcal{O}(2^n n^2)$.

Solved by 12 teams
First solution after 27 minutes

Image Compression

- Find the combination of PNG filters that maximizes the frequency of any byte, and return the byte and its frequency.

Solved by 11 teams
First solution after 29 minutes

Image Compression

- Find the combination of PNG filters that maximizes the frequency of any byte, and return the byte and its frequency.
- We can do this greedily: Compute all 4 filters for a row, and for each byte, store only the maximum frequency we saw from the filters computed.
- Then sum up the maximums for each byte, and pick the one with highest frequency.

Solved by 11 teams
First solution after 29 minutes

Image Compression

- Find the combination of PNG filters that maximizes the frequency of any byte, and return the byte and its frequency.
- We can do this greedily: Compute all 4 filters for a row, and for each byte, store only the maximum frequency we saw from the filters computed.
- Then sum up the maximums for each byte, and pick the one with highest frequency.
- **NB:** `(Orig(x) - floor((Orig(a)+Orig(b))/2)) % 256` is not sufficient! You must do modulo after each addition and subtraction:
  `(Orig(x) - floor(((Orig(a)+Orig(b)) % 256) / 2)) % 256`

Solved by 11 teams
First solution after 29 minutes

- Find the personality types of each candidate

- Find the personality types of each candidate
- Note: The problem statement explicitly tells you there's exactly one solution, so no need to take into consideration no or multiple valid answers

- Find the personality types of each candidate
- Note: The problem statement explicitly tells you there's exactly one solution, so no need to take into consideration no or multiple valid answers
- Since $3^7 = 2187$ ($7 =$ max number of candidates), we can try all character type combinations

To check whether the character type combination is valid, all of these tests must be checked:

- A truther does not lie (speak a false statement)
- A fabulist never tells the truth
- The first utterance of a charlatan must be a truth
- A charlatan will not speak a truth after lying
- A charlatan will speak at least one truth and one lie

Note that only a charlatan must speak, truthers and fabulists may stay silent.

Solved by 7 teams
First solution after 47 minutes

Lemonade

Authors: Nils Barlaug, Sondre Sortland

- Given M possible attempts and a N litre tank, maximise the amount of lemons worth of juice you can fetch based upon some formulas.

Solved by 7 teams
First solution after 55 minutes

Lemonade

Authors: Nils Barlaug, Sondre Sortland

- Given M possible attempts and a N litre tank, maximise the amount of lemons worth of juice you can fetch based upon some formulas.
- Lemons worth of juice **is not** the same as maximising volume!

Solved by 7 teams
First solution after 55 minutes

Lemonade

Authors: Nils Barlaug, Sondre Sortland

- Given M possible attempts and a N litre tank, maximise the amount of lemons worth of juice you can fetch based upon some formulas.
- Lemons worth of juice **is not** the same as maximising volume!
- "Triple" dynamic programming:
  - Iterate over the M possible attempts
  - Keep track of how much volume you have left in the tank *and* v_i^*, the lemons worth of juice in the dog's tank
  - Maximize the lemon's worth of juice you have
- $\mathcal{O}(MNc) = \mathcal{O}(n^3)$

Solved by 7 teams
First solution after 55 minutes

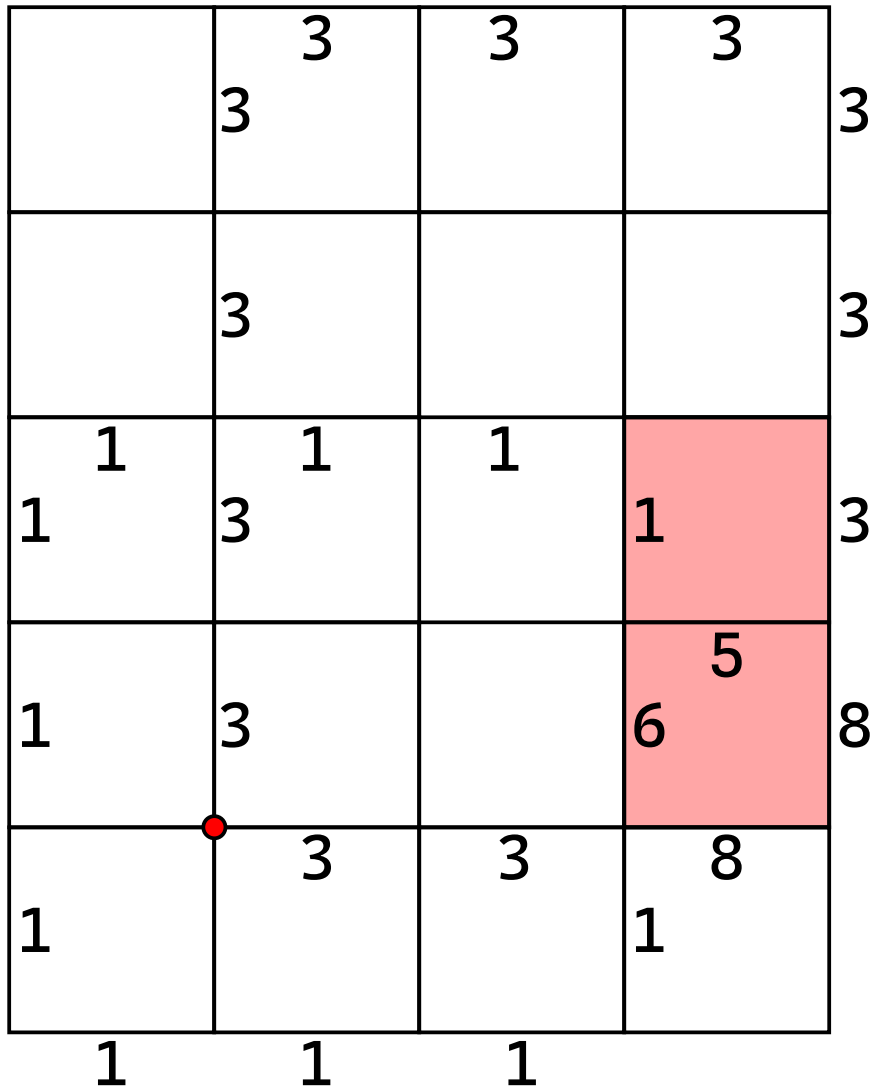
Garbage Tracking

Author: Jean Niklas L'orange

- Garbage has been dumped out over a city grid's streets by robots going in rectangles. Your task is to print the amount of garbage in multiple rectangular sections of the grid.
- The grid can be big! $N = W \times H = 4500 \times 4500 \approx 20$ million
- But first: How do we represent the grid? It's easiest with two grids, with vertical street segments in one grid, and horizontal street segments in another:

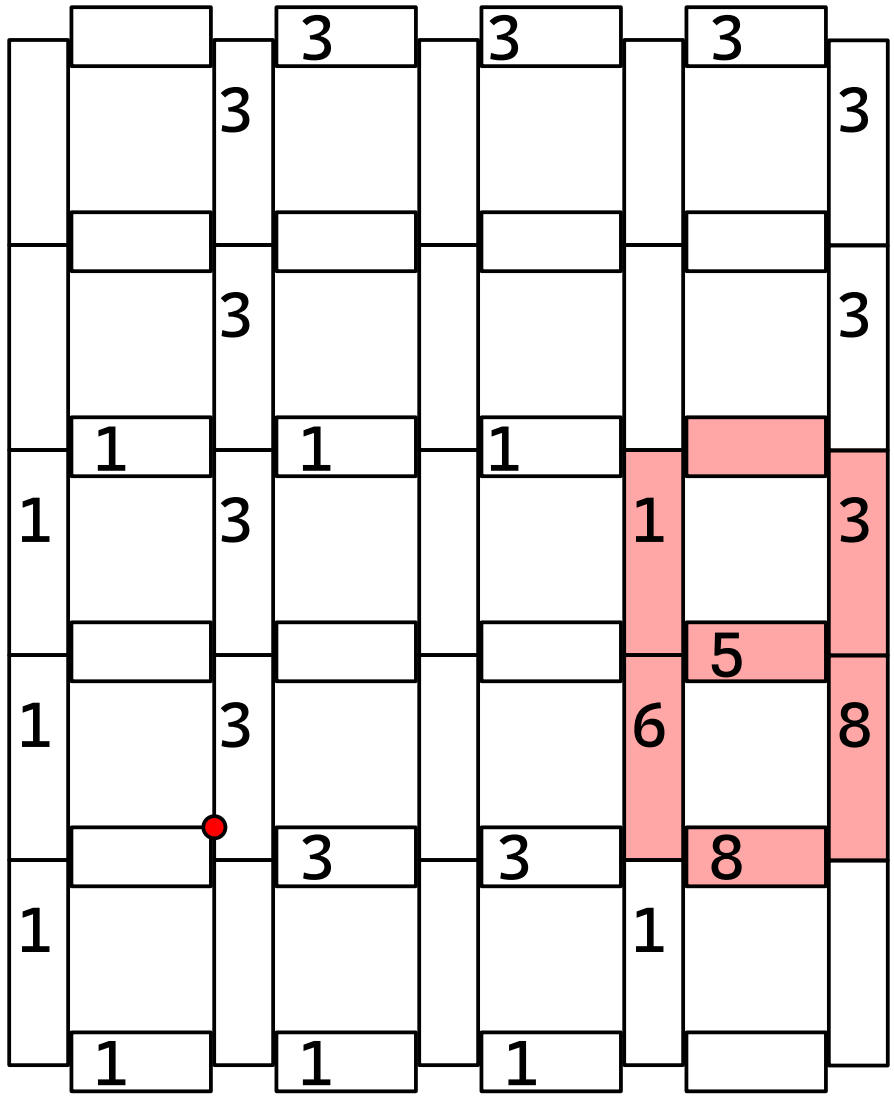
Garbage Tracking

Author: Jean Niklas L'orange



Garbage Tracking

Author: Jean Niklas L'orange

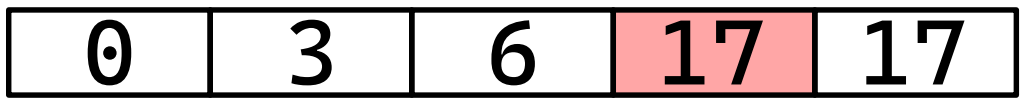


- All operations and queries must be very fast, as there are many of them and the grid is big
- Note: First all operations happen, then all queries happen.
  $\Longrightarrow$ We can do an expensive processing step between the operations and queries

Values stored as number of elements at particular point:

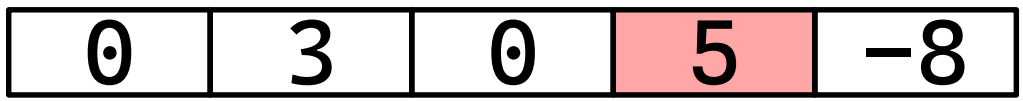
- Update range from a to b: $\mathcal{O}(b - a)$
- Fetch range from a to b: $\mathcal{O}(b - a)$

Values stored as cumulative sum of values:



- Update range from a to b: $\mathcal{O}(n - a)$
- Fetch range from a to b: $\mathcal{O}(1)$

Delta encoding: Values stored as $arr[i] - arr[i-1]$ (like Sub from Image Compression):



- Update range from a to b: $\mathcal{O}(1)$
- Fetch range from a to b: $\mathcal{O}(b)$

Garbage Tracking

Author: Jean Niklas L'orange

Action plan:

- Use delta encoding during insertion of values: $\mathcal{O}(1)$ per operation
- Translate grid into cumulative sum of values: $\mathcal{O}(W \times H)$ once
- Use cumulative sum of values for queries: $\mathcal{O}(1)$ per operation
- Gives in total $\mathcal{O}(W \times H + R + Q)$ running time

Garbage Tracking

Author: Jean Niklas L'orange

- A little trickier in practice, as we're dealing with 2D arrays and not 1D arrays
- Use inclusion-exclusion pricinciple to get the sum of a rectangle from cumulative 2D array
- Easy to get off-by-one errors here. It's not stupid to make a naïve implementation and some sample grids first, and then compare it with the optimised version.

Solved by 7 teams
First solution after 63 minutes

- Problem statements says emitters must send a unique frequency, and the reflectors can only reflect one frequency, so there is a one-to-one pairing.

Solved by 4 teams
First solution after 40 minutes

Authors: Nils Barlaug, Sondre Sortland

- Problem statements says emitters must send a unique frequency, and the reflectors can only reflect one frequency, so there is a one-to-one pairing.
- one-to-one pairings typically reflect bipartite matching, and that's what you're supposed to do.

Solved by 4 teams
First solution after 40 minutes

Authors: Nils Barlaug, Sondre Sortland

- Problem statements says emitters must send a unique frequency, and the reflectors can only reflect one frequency, so there is a one-to-one pairing.
- one-to-one pairings typically reflect bipartite matching, and that's what you're supposed to do.
- Use binary search to find the minimum angle that allows all reflectors to be mathed with an emitter.
- (The geometry part is probably the hardest piece of the puzzle for most)

Solved by 4 teams
First solution after 40 minutes

Captured by Aliens

Author: Sander Land

- Looks and sounds more daunting than it really is!

Solved by 3 teams
First solution after 194 minutes

Captured by Aliens

Author: Sander Land

- Looks and sounds more daunting than it really is!
- Simulation: Place stones on the board and check if neighbouring groups of stones are captured by the definition

Solved by 3 teams
First solution after 194 minutes

Captured by Aliens

Author: Sander Land

- Looks and sounds more daunting than it really is!
- Simulation: Place stones on the board and check if neighbouring groups of stones are captured by the definition
- It's a bit too slow to manually check the edges of a group each time, so use sets to denote stones and empty adjacent edges. Use these to update neighbouring groups whenever a stone is placed and groups are captured.
- No need to do clever logic to merge the sets: Just take the union of the sets and make a new bigger group.

Solved by 3 teams
First solution after 194 minutes

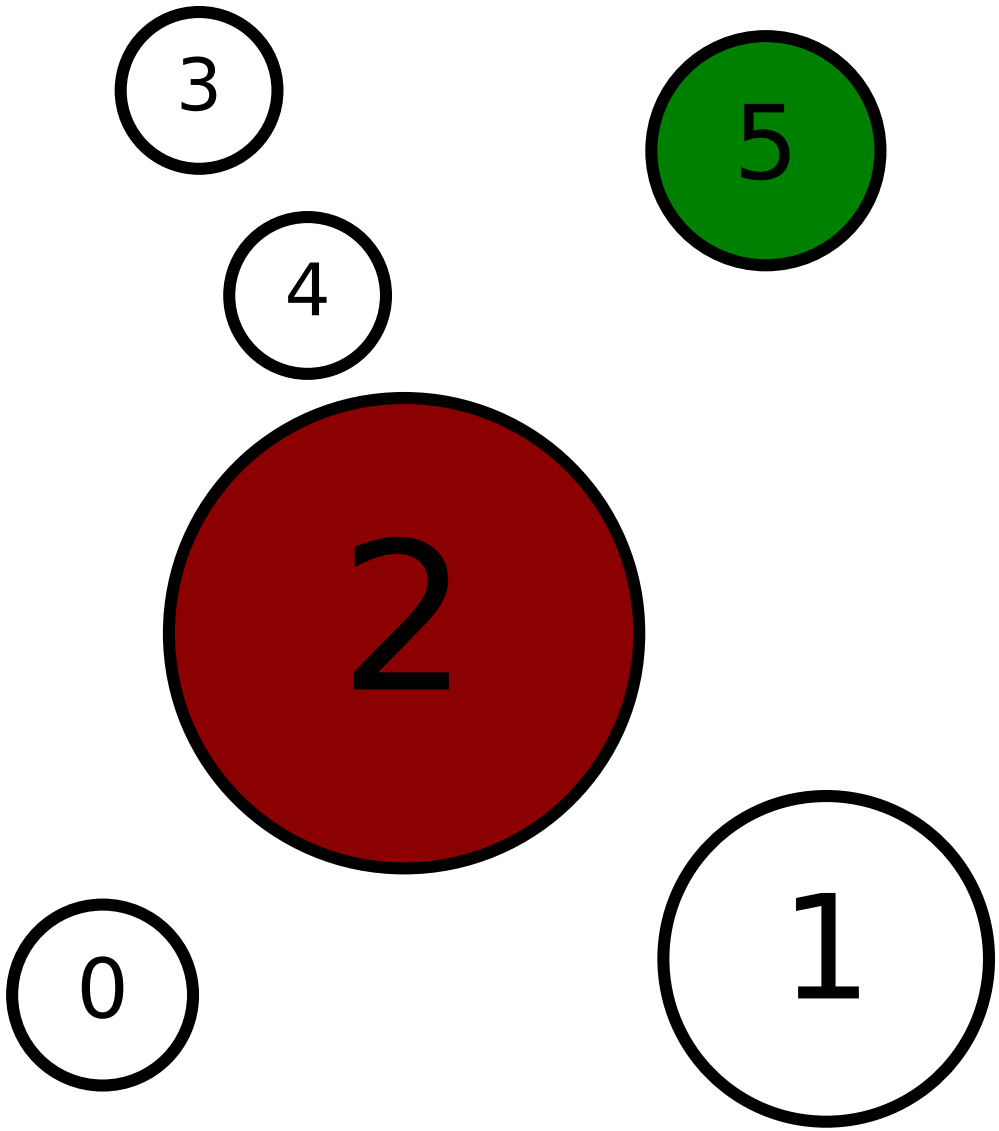
Helpful Rotations

Author: Jean Niklas L'orange

- Shortest path in 2d-space over rotating and non-rotating circles
- For the most part dealing with a lot of small independent geometry pieces you glue together
- When all the geometry is done, you can construct a graph with nodes and edges you can perform Dijksta on

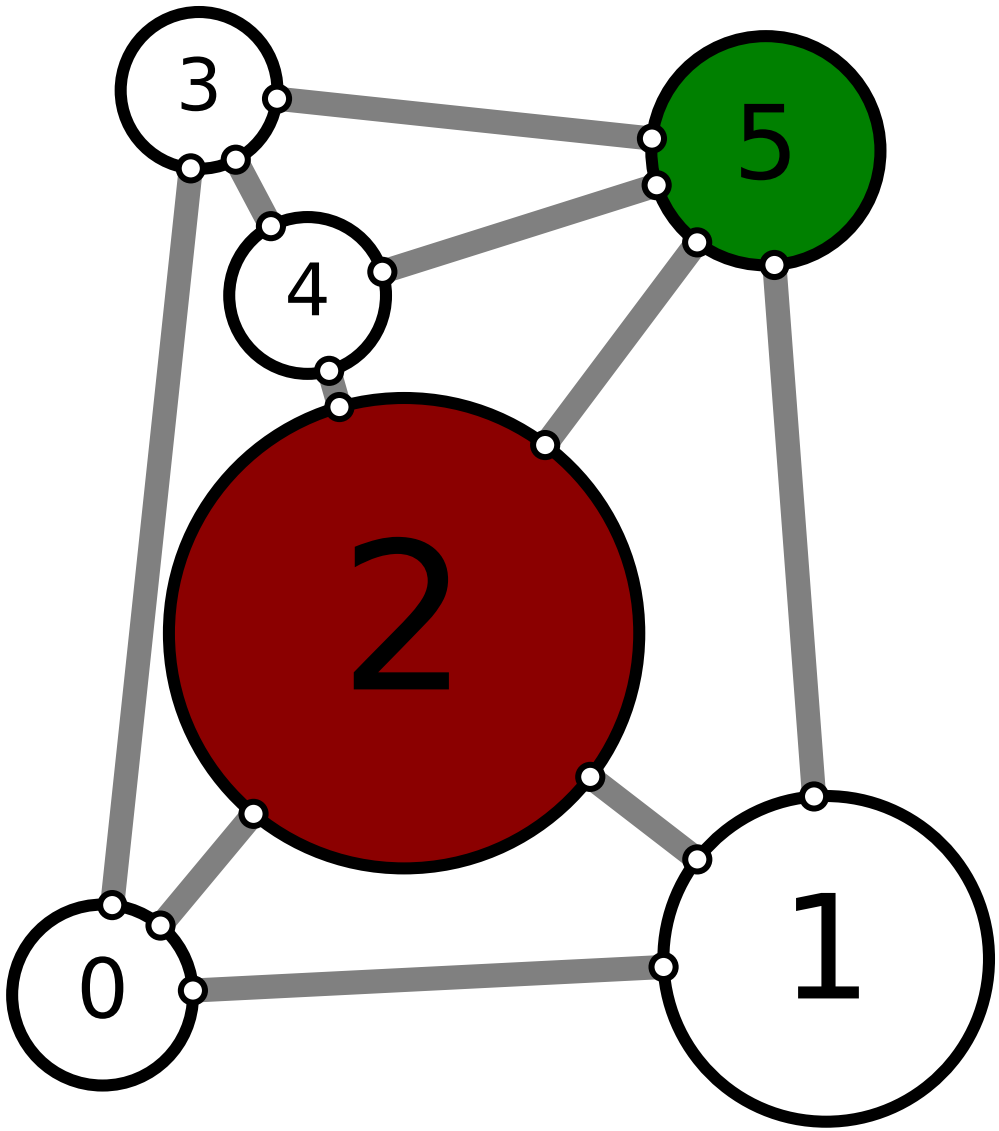
Helpful Rotations

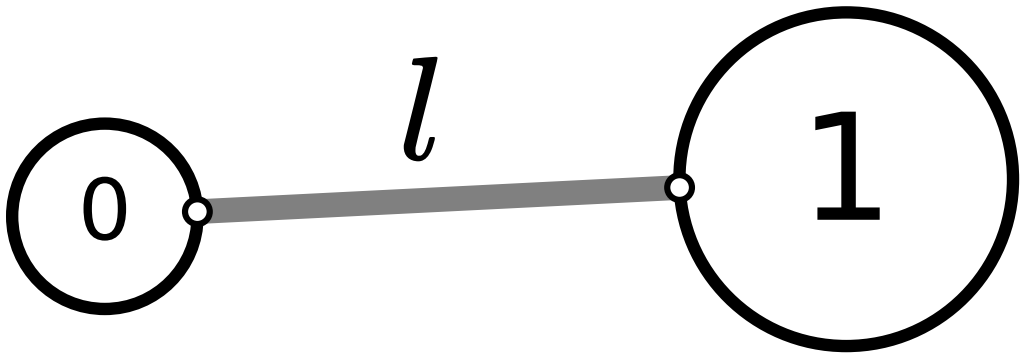
Author: Jean Niklas L'orange



Helpful Rotations

Author: Jean Niklas L'orange





- The shortest distance from disc $d1$ to disc $d2$ is the euclidean distance from their surface:

$$l = \sqrt{(d1.x - d2.x)^2 + (d1.y - d2.y)^2} - d1.r - d2.r$$

where r is the radius of the disc.

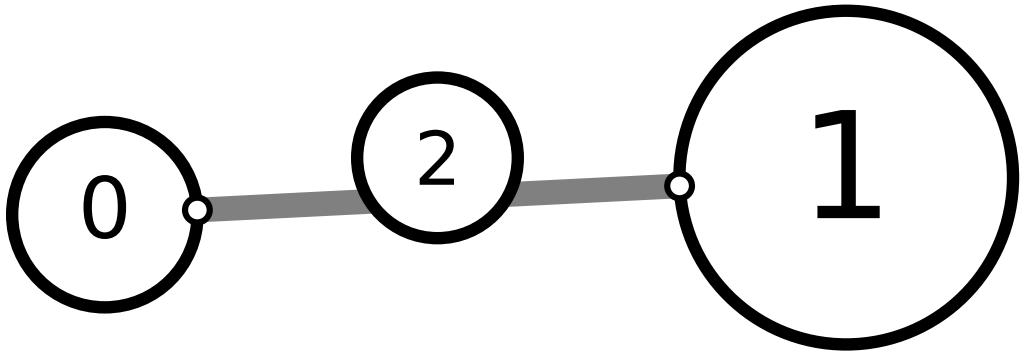
- The fastest way to another disc is if the spaceship accelerates half the way to another disc, then decelerate the remaining half. The formula for distance travelled given constant acceleration a, $v_0 = 0$ and t time passed is $d = a \times \frac{t^2}{2}$
- Solving for t gives us $t = \sqrt{\frac{2d}{a}}$ and since d in this case is half the length ($d = \frac{l}{2}$), we end up with $t = \sqrt{\frac{l}{a}}$ for half the distance, and

$$t = 2\sqrt{\frac{l}{a}}$$

the entire way.

Helpful Rotations

Author: Jean Niklas L'orange



- A path from disc $d1$ and disc $d2$ can only be used if no other disc intersects that line segment
- With at most 175 discs, you check that none of the other discs intersects naïvely by trying them all
- It's possible to deduce the formula without Internet through some thinking and by solving a quadratic formula, although having a "line segment intersects circle" formula readily available makes this problem easier

Helpful Rotations

Author: Jean Niklas L'orange

- Limits sounds complicated, but essentially boils down to: Discs won't overlap and you don't have to worry about floating point issues

Solved by 3 teams
First solution after 109 minutes

- Create asteroids by copying other asteroids' mineral veins and scaling them. Also remove some mineral veins now and then. Finally, output mineral data (total amount + biggest).

- Create asteroids by copying other asteroids' mineral veins and scaling them. Also remove some mineral veins now and then. Finally, output mineral data (total amount + biggest).
- Can't deep copy the contents of asteroids due to the exponential nature of the problem.

- Create asteroids by copying other asteroids' mineral veins and scaling them. Also remove some mineral veins now and then. Finally, output mineral data (total amount + biggest).
- Can't deep copy the contents of asteroids due to the exponential nature of the problem.
- Can't reference the other asteroid either, as it may be mined later.

Asteroid Mining

Author: Jean Niklas L'orange

- Solution: Use a purely functional heap with efficient merge, insert and delete. In practice, any kind will do.

Solved by 2 team
First solution after 210 minutes

Asteroid Mining

Author: Jean Niklas L'orange

- Solution: Use a purely functional heap with efficient merge, insert and delete. In practice, any kind will do.
- Leftist Heaps are probably the easiest to implement, and has $\mathcal{O}(\log n)$ running time for all necessary operations.

Solved by 2 team
First solution after 210 minutes

Asteroid Mining

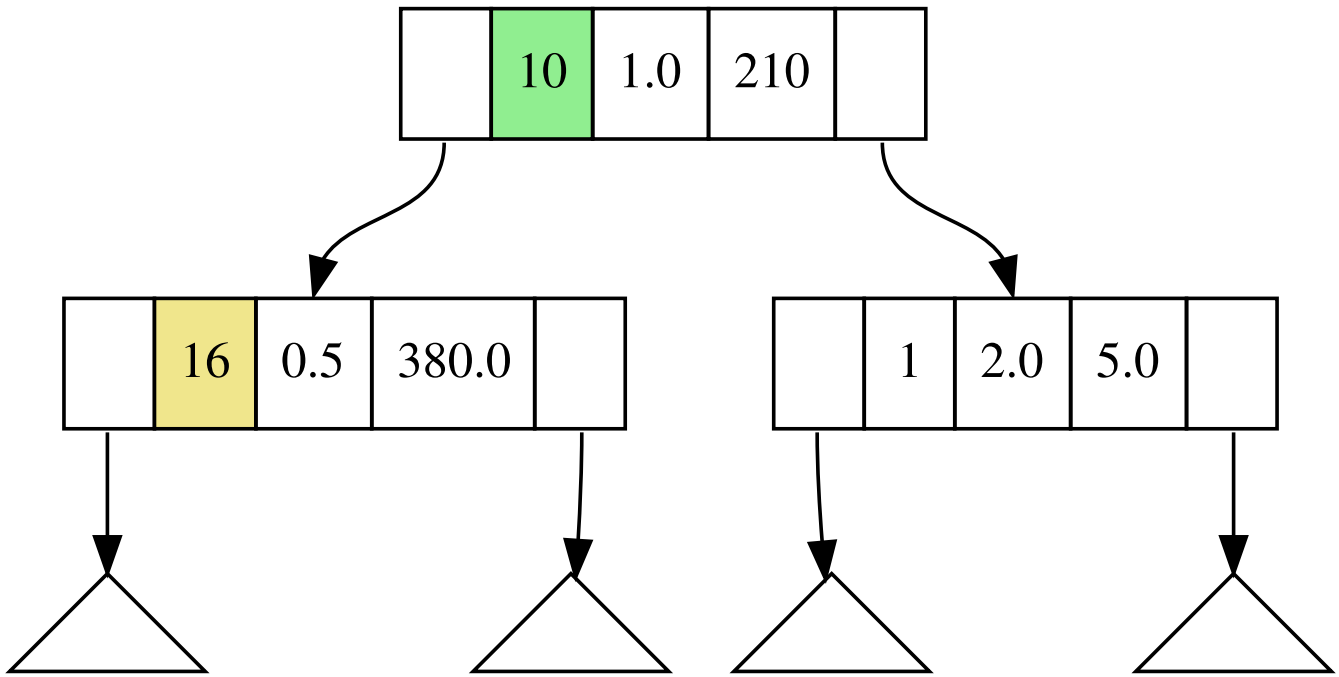
Author: Jean Niklas L'orange

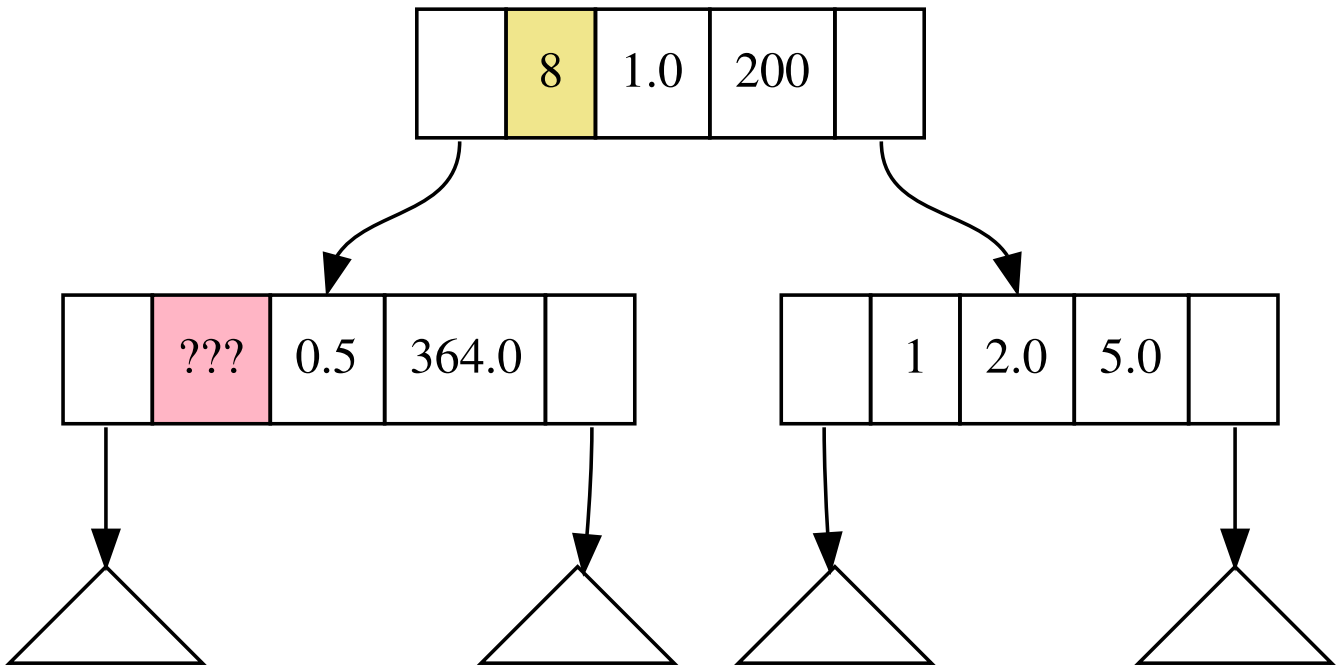
- Solution: Use a purely functional heap with efficient merge, insert and delete. In practice, any kind will do.
- Leftist Heaps are probably the easiest to implement, and has $\mathcal{O}(\log n)$ running time for all necessary operations.
- Add a scale factor to each heap node to delay scaling, and a total sum to make the final query quick to compute.
- Runs in $\mathcal{O}(\log(2^A)E) = \mathcal{O}(AE)$

Solved by 2 team
First solution after 210 minutes

Asteroid Mining

Author: Jean Niklas L'orange




8
1.0
200
???
0.5
364.0
1
2.0
5.0