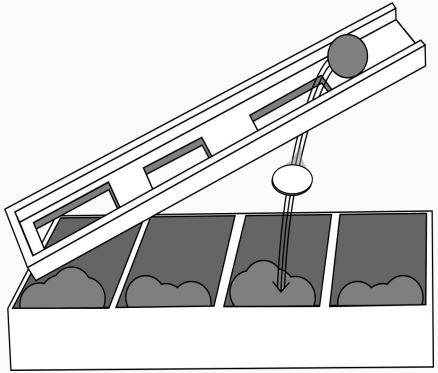
UKIEPC 2019

Summary and solution outlines

Problem Solutions



Auto Accountant

24 correct • solved at: **01:02** by
AmaTRINciana
University of Cambridge

Author: **Robin Lee**

Overview

- We represent a coin with coords (X,Y) pair. A coin falls into a slot (U,V) if X≤U and Y≥V.
- For each coin, find the first slot in the list that matches and add its index to the answer.

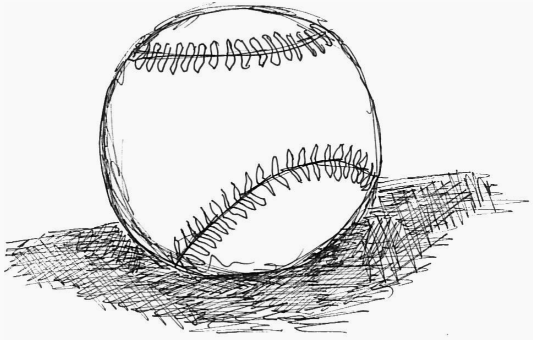
Automatic Accountant - Solution

Techniques

- Segment trees
- KD trees

Algorithm

- Keep a **segment tree** mapping for one axis:
  - For all slots with thickness >= T,
  - Which one has the lowest index? (min-segment-tree)
  - Initially this tree is empty
- Sort the coins and slots along the other axis
  - Iterate through both in parallel, inserting slots as their trigger masses become eligible for the current coin.
  - Use the tree to find the slot with the smallest index, out of those with the right mass range.
- Alternatively, use a KD / quad tree



Ballpark

174 correct • solved at: **00:03** by

Ananas
University of Cambridge

Author: **Jim Grimmett**

Overview

- Estimate a number to one significant figure (exactly one nonzero digit).
- The number fits inside a 64-bit integer
    - (c++: **int64_t**)
    - (java: **long**)
    - (python: **number**)

Ballpark Estimate - Solution

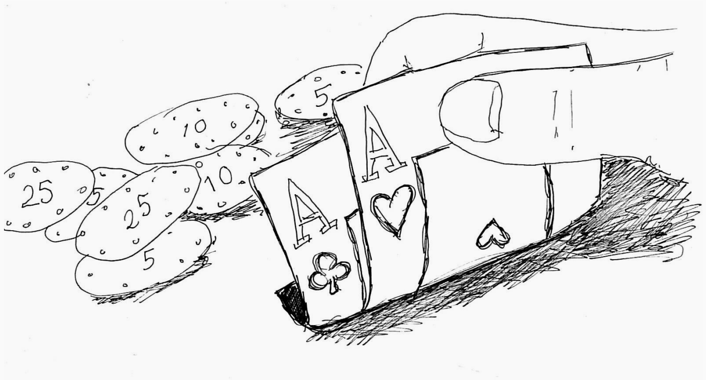
Techniques

- Logarithms
- Rounding

Algorithm

- If we reduce the number to {x}.{abcdefg} where x is a single-digit number, we can just round it and add zeroes back on later. We just need to make sure to keep the extra information after the decimal point.
    - while (number >= 10) { tens++, number /= 10.0; }
    - number = int(round(number))
    - while (tens > 0) {tens--, number *= 10; }

- Or (since only the first two digits matter):
    - int(round(int(s[0:2]) / 10.0)) * (10**(len(s)-3))



Crooked Dealing

81 correct • solved at: **00:20** by

Treeniceratops
University of Cambridge

Author: **Robin Lee**

Overview

- Partition some numbers into as many groups of K as possible,
- **But** make sure the name number never shows up in the same partition twice.

Crooked Dealing- Solution

Techniques

- Greedy algorithms
- Priority queues

Algorithm

- Use a hashmap (or Python's Counter class) to get the frequency of all the cards. It's always best to try and get rid of the most frequent card as fast as possible.
- Put the cards into a priority queue ordered by frequency.
- While the queue has enough elements to make a hand:
    - Pop the largest K items from the queue
    - Add the values to the answer
    - Reduce the frequencies by one
    - Reinsert the items and new frequencies into the queue
        - They may not have the same ordering in the queue afterwards.
- Or, binary search on the answer X, lay the numbers out into a grid with X columns, and the answer is the columns of the grid.

Dome

122 correct • solved at: **00:11** by

BigBoggerBoys2:ElectricBoogaloo
Dublin City University

Author: **Jim Grimmett**

Overview

- There are some points in 3D space
- We have a dome sited at the origin
- How big do we have to make the dome to capture K or more of the points?

Dome Construction- Solution

Techniques

- Sorting
- Geometry

Algorithm

- The actual positions of the points don’t matter, just how far they are from the origin. Map the points to hypot(x,y,z) or hypot(hypot(x,y),z) if your programming language doesn’t take 3 arguments.
- Now sort them. This will put the closest K distances as the first K elements of the array!
    - So now you can just print the Kth element.

- Or: binary search on the answer (a very versatile algorithm) and count how many points match to decide to go lower/higher.

Estate Agent

6 correct • solved at: **01:15** by

Treeniceratops
University of Cambridge

Author:

Bjarki Ágúst Guðmundsson

Overview

- Some people want to buy each others' houses. We want to earn money.
- What's the largest possible sum of transactions we can make?

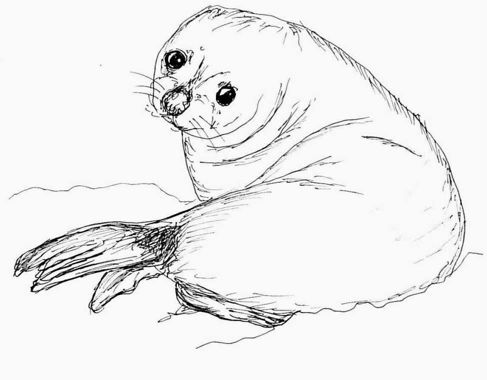
Estate Agent - Solution

Techniques

- Bipartite graphs
- Hungarian algorithm

Algorithm

- Make a graph where people are vertices, and so are houses. Make an edge between a person and a house if they want to buy it and assign the offer value as the weight.
  - Crucially, **also** make an edge between a person and their own house with a zero weight. This is the default case.
- Now we have another bipartite matching problem.
  - The graph is weighted, so we need to use the Hungarian algorithm or a minimum-cost-maximum-flow (MCMF) algorithm.
  - Plug in and play after setting up the appropriate graph.



Feeding Seals

99 correct • solved at: **00:07** by

Treevial
University of Cambridge

Author: **Ian Pratt-Hartmann**

Overview

- We can give a person 2 buckets if their combined weight is less than or equal to some constant C.
  - But if we can't do that, or don't want to, we can give them just one bucket.
- To carry N buckets of various weights, how many people do we need?

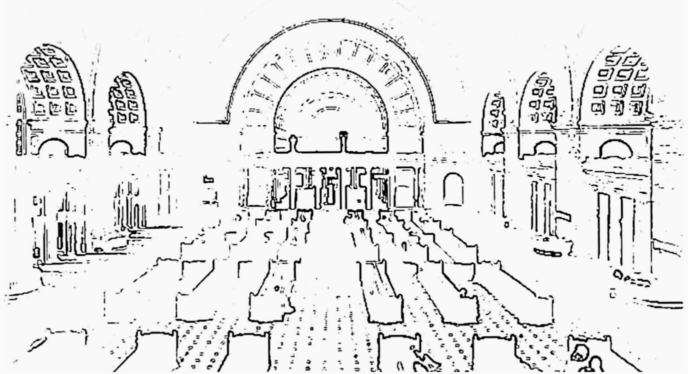
Feeding Seals - Solution

Techniques

- Sorting
- Two pointers

Algorithm

- This is a class of problem called "two pointers". If we sort all of the weights, we can solve it with a kind of recursive argument:
  - If anything is going to be paired up, it makes sense to use the smallest item as part of a pair.
  - We should also use as big an item as possible with the smallest item.
    - If this can be the largest item, that's the best option. We throw both the start and end of the array away.
    - Otherwise, we can **never** pair the largest item, so we throw it away.
  - Use two pointers into the ends of the array (or a deque) to implement this efficiently.



Grand Central Station

7 correct • solved at: **02:19** by

Treeniceratops

University of Cambridge

Author: **Robin Lee**

Overview

- We have an unrooted tree.
- We have some anonymous nodes in the tree connected to each other.
- How many of the nodes are functionally the same (isomorphic)?

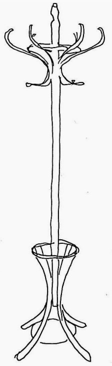
Grand Central Station - Solution

Techniques

- Tree centroids
- Isomorphism
- Hashing

Algorithm

- We need a canonical label for each node of the tree.
  - One way is to make a hash for a node, by taking the hashes of all the other nodes around it and hashing them into one super-hash
  - Sounds impossible but can be done by excluding one neighbour node at a time.
- Another way is to root the tree at its centroid- found by taking the longest path in the tree and looking for the middle node(s) in this path.
  - Then each node can have a label, and nodes with the same list of child labels can have the same label.
  - If a node has two child labels, merge them together and count them. Time O(N).



Hat Stand

27 correct • solved at: **00:36** by
Treevial
University of Cambridge

Author: **Robin Lee**

Overview

- We have a unique kind of cache for hats. The last-used item is put in the place of the next-used item.
- What is the best way of optimising this cache?

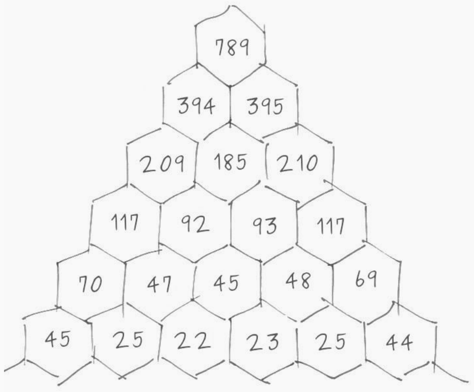
Hat Stand - Solution

Techniques

- Simulation
- Sorting

Algorithm

- Let's say we already picked an ordering of the hats and simulated it. What would the cost be?
  - For each starting hook, count the number of accesses and multiply by its index.
  - For a given hat: the number of accesses for the hook the hat starts on is constant, but we can change the index.
  - Let's count the number of accesses in a "default" permutation, and reorder starting from the most accessed items to reduce cost.
- Key insight is to forget about the ordering to begin, and only apply it when it starts to matter.



Integral Pyramid

78 correct • solved at: **00:10** by

When all else fails take a nap
University of Cambridge

Author: **Robin Lee**

Overview

- A pyramid is made by adding numbers on lower rows together.
- We want to make a given number at the top. What should the numbers at the bottom be?

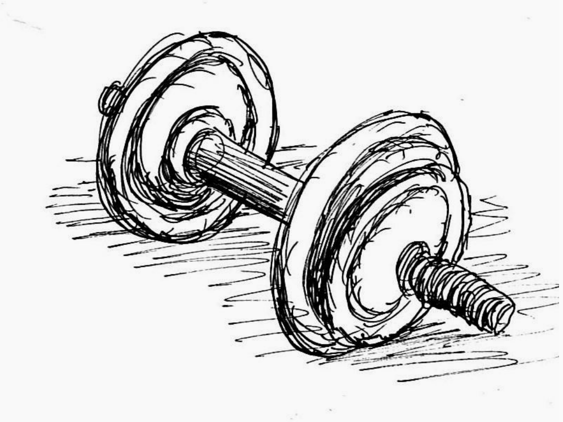
Integral Pyramid - Solution

Techniques

- Dynamic programming
- Cheekiness

Algorithm

- Start by just putting all 1s in the bottom row.
  - This gives a sum of 2^(n-1) at the pinnacle.
- Now, because there's only one way for the first and last items to "contribute" to the final score, we can make up the difference in column 0 by adding to it.
  - As long as we make sure this addition is non-negative. If not, the test case is impossible.
- Nicer ways are possible too, but why bother?



Jammed Gym

45 correct • solved at: **00:25** by
Kvalitní Slovenskí Programátori
University of Cambridge

Author: **Robin Lee**

Overview

- Find a shortest path where each node has multiple locations.

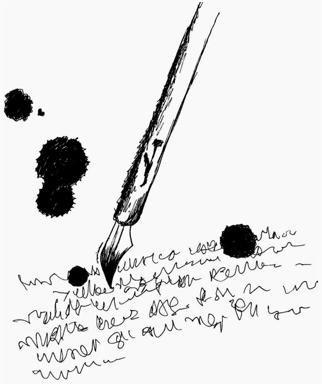
Jammed Gym - Solution

Techniques

- Dijkstra's algorithm
- Dynamic programming

Algorithm

- Really, nodes of the same kind are **not** the same, we just need to go to any of them at some time T.
- So we can make a table of cost_to_visit[T][NodeId] and only fill it in for the relevant kinds of node at time T.
    - Iterate through T in increasing order and do an all-pairs comparison to find if:
        - Station at T is valid to leave from
        - Station at T+1 is valid to go to.
- Read off the minimum number in row T of the matrix at the end.



Knocked Ink

2 correct • solved at: **04:13** by

Treeniceratops
University of Cambridge

Author: **Robin Lee**

Overview

- Ink is spreading across a page in circles.
- Some ink blots start earlier, others later.
- How long until the total area is A?

Knocked Ink - Solution

Techniques

- Circle intersection
- Line integrals
- Green's theorem
- Binary search
- Pain tolerance

Algorithm

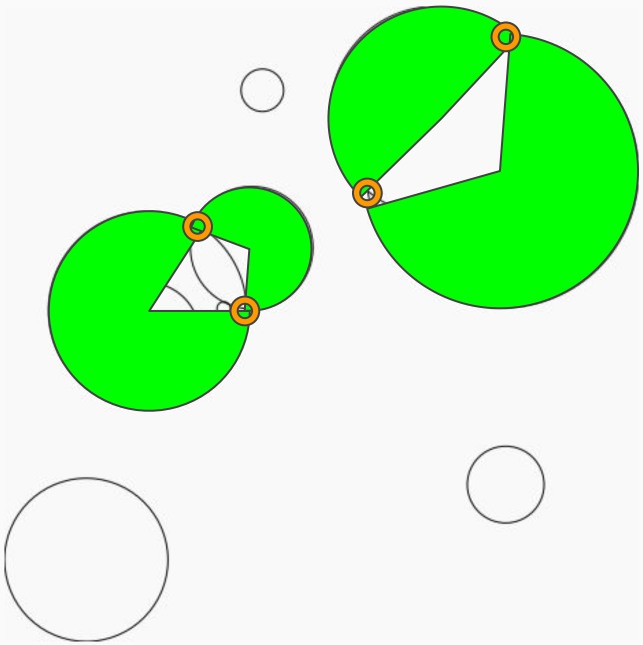
- The spreading out of ink is the easy bit- area covered only increases, so we can run binary search (100+ iterations is plenty).
- Now we have to check the area of union of the blots. This is not as easy as it sounds.
  - Some areas are just covered by one or two blots, other areas can be covered by dozens of blots with circle edges all over the place.
  - If we can describe the intersecting circles as one continuous polyline, our job is much easier- when we can describe a curve mathematically, we can probably integrate it mathematically too.
- Let's start by figuring out which arcs are on the border

Knocked Ink - Solution

Techniques

- Circle intersection
- Line integrals
- Green’s theorem
- Binary search
- Pain tolerance

Algorithm



Knocked Ink - Solution

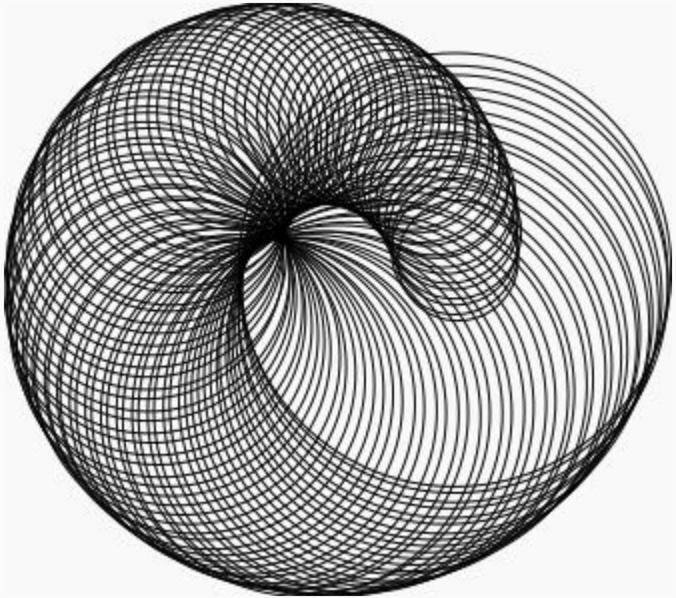
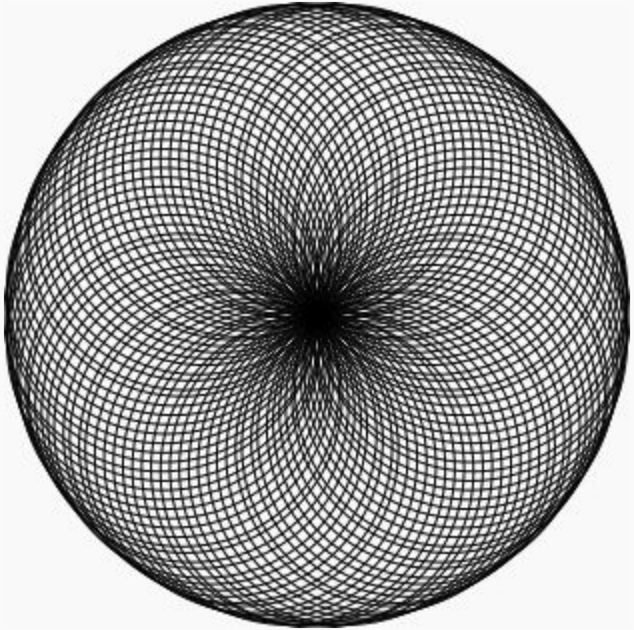
Techniques

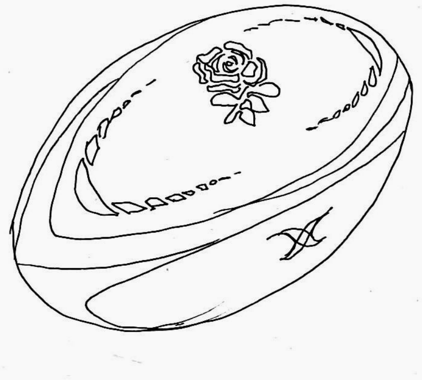
- Circle intersection
- Line integrals
- Green’s theorem
- Binary search
- Pain tolerance

Algorithm

But… Why not use a spatial data structure?

- Let’s take a look.



Low Effort League

12 correct • solved at: **01:05** by
??!
University of Cambridge

Author: **Robin Lee**

Overview

- How many games do you have to rig/modify to win a tournament?
- Specifically, how do you minimise total cost to win if cost to win one game is the square of the difference in skill?

Low Effort League - Solution

Techniques

- Dynamic programming

Algorithm

- Similar to Jammed Gym- dynamic programming
    - Cost to have team X in round R = cost[X][R]. This can be calculated by finding all teams T in the adjacent bracket in round R and comparing against cost[T][R-1].
    - There are X*R cells = R*2^R cells. This is a lot, but not too many to make it slow.
    - Here, just read off the value of cost[1][R] for the answer.

Mosaic

8 correct • solved at: **01:47** by

Treeniceratops
University of Cambridge

Author: **Robin Lee**

Overview

- Remove some rows from a rectangular array to make every value in the array show up equally often.

Mosaic - Solution

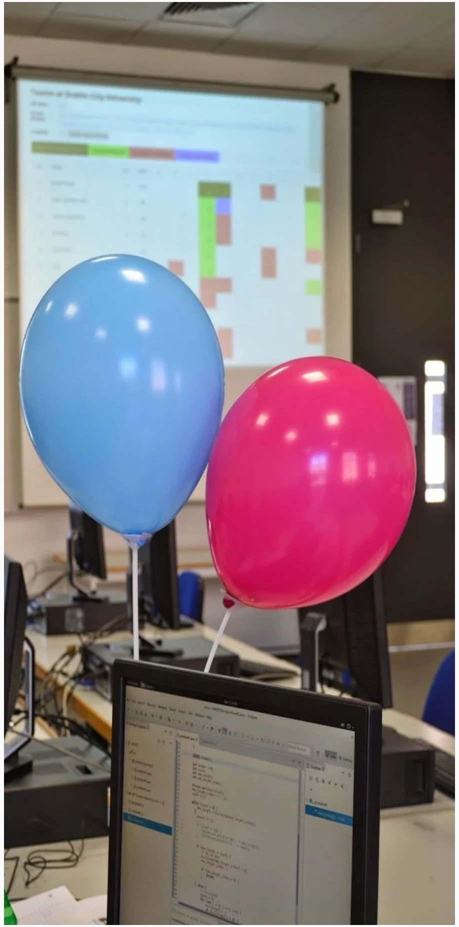
Techniques

- Meet in the middle
- Hashing

Algorithm

- Meet in the middle- break 2^40 worth of brute force into 2^20 x 2
- Find two “half solutions” which cancel each other out, for example 2xA+1xB in one, and 2xA+3xB in the other.
- This is fast enough if the arrays are small,
  - But the arrays are very large
  - So make a hash function that still supports adding together and subtracting values in aggregate without recalculating the whole thing
    - For safety, make several such hash functions in case any one is weak, and bundle them together.






Questions?

Or comments?

Final Standings

http://domjudge.bath.ac.uk/