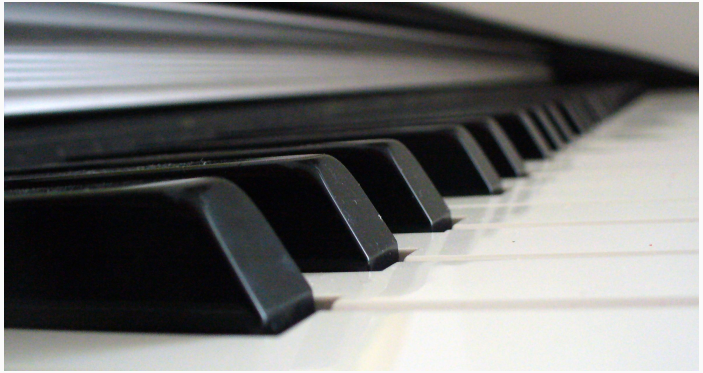
UKIEPC 2018

Summary and solution outlines

Problem Solutions



Analogue Cluster

67 correct • solved at: **00:10** by
Spaghetti Coders (Cambridge)

Authors:
Nir Oren and **Robin Lee**

Overview

- We have a simple graph with vertices having different "types"
- Make every pair of vertices connected by an edge the same "type"
- Change the minimum possible number of vertices.

Analogue ClusterAnalogue Cluster - Solution

Techniques

- Connected Components

Algorithm

- Find the connected components
- In each component, find the most popular width
- Modify all the others to fix

Bus Logic

93 correct • solved at: **00:09** by

Spartans (Cambridge)

Author: **Robin Lee**

Overview

- We have a set number of stops and a list of bus routes showing which stop each bus visits.
- When a bus finishes a route, it drives back to the first stop in the route and starts again.
- Given a starting stop, find which busses visit it, and then count the number of distinct stops that set of busses visits.

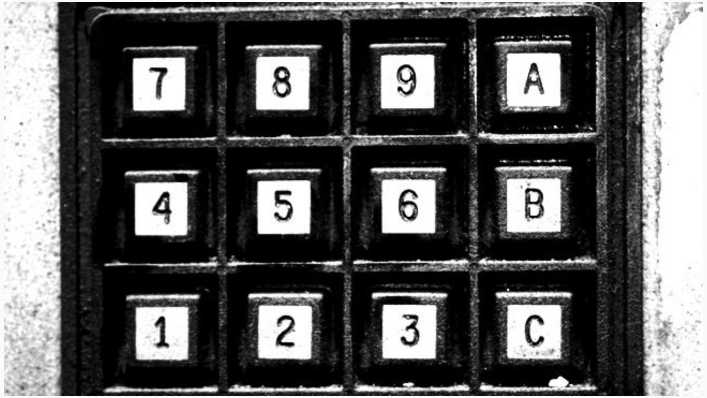
Bus Logic - Solution

Techniques

- Bitmasks
- Logic

Algorithm

- Represent the bus routes as integer bitmasks
  - 0 = bus does not visit this stop
  - 1 = bus does visit this stop
- We can change to any bus **b** if it overlaps with our starting bus **a**
  - (**a** AND **b**) ≠ 0
- We can reach any stop on a bus route **b** that is changeable-to
  - **answer** = (**answer** OR **b**) IF (**a** AND **b**) ≠ 0
- Count the number of bits and print!
  - bits(x) ← **if** bits(x) == 0 **then** 0 **else** bits(x%2) + bits(x/2)
  - bits(x) ← __builtin_popcount(x) *in c++*
  - bits(x) ← bin(x).count('1') *in python*



Code Word

35 correct • solved at: **00:22** by
Triniceratops (Cambridge)

Author: **Robin Lee**

Overview

- Output the number of ways of typing in a code on a number grid,
- **Excluding** any ways in which two keys are too close to each other,
- **Modulo** a large number.

Code Word - Solution

Techniques

- Dynamic programming
-

Algorithm

- Modulo is not a problem if we only add and subtract and take the modlus every time.
- Solve by induction on “L”.
  - The valid solution of length L ending in (x,y) are all those of length L-1 except the forbidden ones.
  - Z(r,c,L) is to be the number of combinations of length l ending with (r,c)
- Initialise Z(r,c,1):=1
- For i from 2 to l do
  - # Ideally Z(r,c,i)=sum all Z(r’,c’,i-1): (r’c,’) not forbidden.
  - # But this is too slow, do an incremental version
  - Z(r,c,i) = sum all Z (constant time)
  - Z(r,c,i) -= sum all Z(r’,c’,i-1): (r’,c’) forbidden (much fewer!)

Dynamo Wheel

5 correct • solved at: **01:52** by
Treenity (Cambridge)

Author: **Robin Lee**

Overview

- Buckets are tied to a wheel
  - At 0 degrees, they become full
  - At 180 degrees, they become empty
- Find the angle at which the mechanical moment due to gravity is highest (clockwise)

Dynamo Wheel - Solution

Techniques

- Two pointers
- Geometry
- Rotation matrices

Algorithm

- The centre of mass of the wheel is a fixed value in its own frame of reference, for a given subset of filled buckets.
- Each bucket has two state changes in a 360° cycle:
  - Becomes full at 360-α
  - Becomes empty at 180-α
  - So, there are at most 2n times the centre of mass changes.
- Sort the fill/unfill events and process them in order, updating the centre of mass.
  - Often, the wheel needs to be rotated in between two points to achieve a maximum value
  - Either differentiation or ternary search work for this, but be careful to stay inside the neighbouring angles.

Evenly Divided

14 correct • solved at: **00:57** by
Red (Bath)

Authors: **Ian Pratt-Hartmann**
and **Robin Lee**

Overview

- Put pairs of tall people and short people togetther in parallel lists
- People may not be matched up if one person is the other's "mentor"
  - ie. If they are neighbouring nodes in a tree structure.

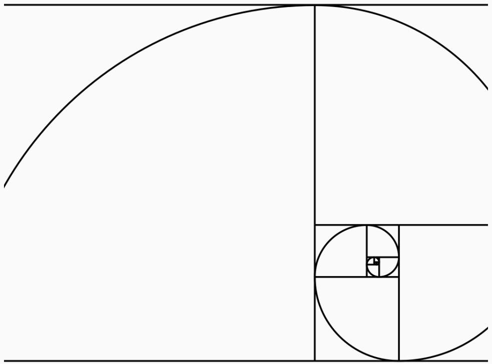
Evenly Divided - Solution

Techniques

- Random matchings
- Bipartite graphs
- Centroids

Algorithm

- This is really a matching problem in reverse.
    - There are up to N-1 matchings we are **not** allowed to make, in the shape of a tree. The rest are fair game.
    - We have a very dense graph in the case where N is large.
    - So a random matching will probably work if it is careful about dealing with the 1 or 2 vertices that can have a high degree.
        - Try 50-100 times; the probability of failing in all of these and there still being a valid matching is minimal.
- There are other, deterministic, algorithms. For example, split the largest connected component at its centroid and only repeatedly pair people across the largest chunks (to avoid conflicts)



Fib Compression

25 correct • solved at: **01:39** by
Me[♠]talci (Cambridge)

Author: **Robin Lee**

Overview

- Given a simple compression algorithm with the number of available symbols growing exponentially with length
  - - Just like most compression algorithms.
- Take each of the N prefixes of a string and compress them separately.
  - What is the compressed length of each one?
  - Note: this is not the same as compressing the whole string and then removing symbols from the end.
    - eg. In a word like ABBCC, the first output is "2", "not "3"!

Fibonacci Compression - Solution

Techniques

- Bookkeeping
- Sorted sets

Algorithm

- Need to maintain sorted frequencies and sorted code lengths and match them up in reverse order. Sorting each time will be too slow.
- Analyse bounds on the size of each "bucket" of codes with identical lengths:
  - n(distinct code lengths) < 40, because fib(40) $\approx 10^8$.
- Because there are few unique lengths, save time by only keeping track of which bucket a symbol goes into, not its exact place.
  - To insert a completely new symbol, add it to the end.
  - To increase a symbol's frequency, repeatedly exchange it with the smallest item in the next bucket until the buckets are in sorted order again.
- Time complexity: $O(N\log^2 N)$.

Garden Variety Vampire

3 correct • solved at: **03:17** by
Triniceratops (Cambridge)

Author: **Jim Grimmett**

Overview

- A number of trees with different shadow sizes
- Three locations to make a shadow-covered path between.
- Calculate if it can be done and output possible or impossible.

Garden Variety Vampire - Solution

Techniques

- Subset enumeration
- Steiner trees
- Geometry
- Power centres

Algorithm

- It's always best to put the biggest circle radius T in the middle.
- We must find coordinates (x,y) for the middle, and a partition (A,B,C) of the other circles st.
  - 2× sum(radii in A) + T <= length((x,y) - v1)
  - 2× sum(radii in B) + T <= length((x,y) - v2)
  - 2× sum(radii in C) + T <= length((x,y) - v3)
- If we know the partition we can check if (x,y) exists by finding one point at which the 3 circles intersect.
  - Ex. 1: find the power centre (always inside an intersection).
  - Ex. 2: take the mean of all pairwise intersection points.
- The number of circles is small, so just try every partition
- Complexity: $O(3^N)$

Hamster Ball

132 correct • solved at: **00:19** by
Treenity (Cambridge)

Author: **Jim Grimmett**

Overview

- A large number of hamster balls of varying sizes
- An amount of tape
- Calculate the largest number of hamster balls than can be sealed shut with the tape.

Hamster Ball - Solution

Techniques

- Sorting
- Greedy Algorithm

Algorithm

- Divide the length of the tape by Pi, then we can use the hamster ball diameter easily.
- Sort the ball sizes smallest first, keeping track of how many of each.
- Starting with the smallest ball size, greedily take away as many balls as we can from the tape until there isn't enough tape left for the next ball or we run out of balls that size, then look at the next size up.

Internet Upload

3 correct • solved at: **02:18** by

Spaghetti Coders (Cambridge)

Author: **Jim Grimmett**

Overview

- We have an amount of data we need to upload.
- We have a list of coffee shops, their upload speeds and when they are open.
- Taking into account time unused when travelling between shops, work out the smallest time you could upload all of the data.

Internet Upload - Solution

Techniques

- Shortest Path
- Line segments

Algorithm

- If we go to a shop, we will either go there until it closes **or** leave at exactly the right time to arrive as a better shop opens.
- Build a graph of potential moves between shops and assign each edge a pair of start and end times.
- Divide each shop into several "mini-shops" across edge start/end boundaries and introduce edges between segments.
- Iterate over the subdivided shops in increasing order of opening time, processing all incoming edges at once.
  - Pick the incoming edge that gives the most data.
  - Update the score for the shop.
  - The optimal answer may be half-way through a visit.
- Time complexity: $O(N^2 \log N)$

Jackpot

50 correct • solved at: **00:47** by
Me[♠]talci (Cambridge)

Author: **Jim Grimmett**

Overview

- A number of closed doors, behind one of which is a prize.

- The prize decreases as doors are opened until the participant decides to try and pick the winning door.

- Expected payout is Remaining Prize Fund * probability of opening the correct door.
- Maximise the Expected Payout.

Jackpot - Solution

Techniques

- Convex functions
- Ternary search
- Differentiation
- Quadratic formula

Algorithm

- The Remaining Prize Fund is a convex function.
- The probability of selecting the correct door is a convex function.
- The Expected Payment function when d doors are open is

$$E(d) = \frac{m-(fd)^2}{n-d}$$

- We can show that this is also convex, so we can use a ternary search to find either a peak, or a max at one end of the range.
- Secondly, if you differentiate the above you get

$$\frac{f^2 d^2 - 2f^2 nd + m}{(d-n)^2}$$

- You can find the maxima by solving the quadratic equation, choose the nearest door, and solve the Expected Payment function

Kings

3 correct • solved at: **01:17** by
Triniceratops(Cambridge)

Author: **Robin Lee**

Overview

- Put chess pieces on the black diagonal of a chessboard.
- They must not collide, they can only move in one direction at once.
- Find the minimum number of moves.

Kings - (Wrong) Solution

Techniques

- Weighted matching
- Exchange lemma
- Greedy algorithms

Algorithm

- Kings will never collide in an optimal solution as they could just swap destinations. Any king can match any square, cost=|dX|+|dY|.
- If N was smaller we could solve with $O(N^3)$ Hungarian algorithm.
- Rephrase the cost function in 1D: $cost(king(x,y), i) = y-x + 2*max(0, i-y, x-i)$ assuming $x<y$.
- For two values of "i" and two kings (x,y) make an exchange argument. Quick casework shows that ordering only matters when there is an intersection, and then we should always be greedy.
- We can make a greedy algorithm:
    - Match the least x coordinate with the closest interval by left endpoint. If there are multiple choose the one with closest right endpoint. Remove the least x coordinate, repeat.

Kings - Solution

Techniques

- Weighted matching

Algorithm

- Rephrase the cost function in 1D: cost(king(x,y), i) = y-x + 2*max(0, i-y, x-i) assuming x<y.
- O(N^3) using the Hungarian algorithm weighted matching algorithm.

Last Word

145 correct • solved at: **00:07** by
AKSLOP-7991 (Cambridge)

Author: **Robin Lee**

Overview

- Substring is called a large number of times on the same string. Print the final string.

Last Word - Solution

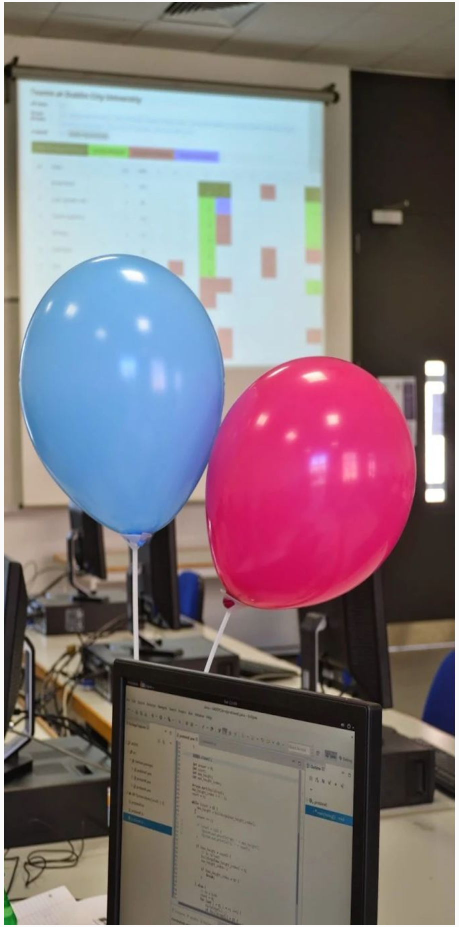
Techniques

- Two pointers

Algorithm

- Instead of copying the whole string every time, advance some pointers from both ends.
    - The start can be the sum of the start indices.
    - The end can be that plus the last length used.
- Then call substring() just once instead of N times.






Questions?

Or comments?

Final Standings

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