UKIEPC Names

Organisers: **Max Wilson**, **James Davenport**, **Rachid Hourizi**

Writers: **Robin Lee, Jim Grimmett**, **James Stanley**

Reviewers: **Ximo Lerma**, **Per Austrin**

Sysadmins: **Neil Francis**, **Matt Richards, Rob Perkins**

Illustrator: **Lisa Abose**

Problem Solutions

E- Ed’s Problem

Author: **Jim**

Overview

A number of tax bands, each with a certain tax percentage.

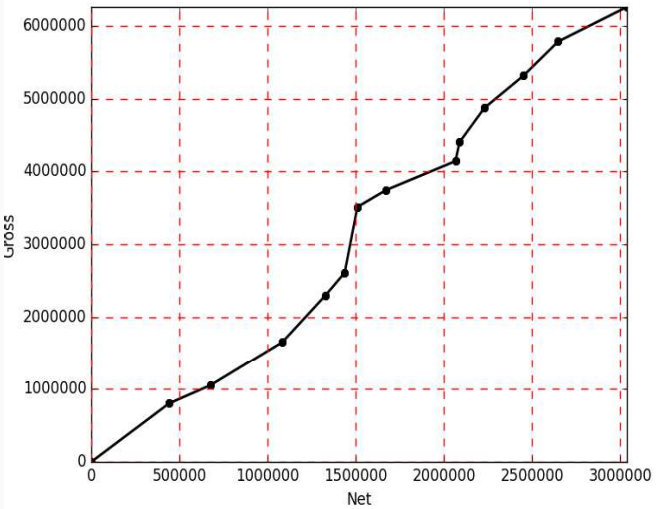
A number of friends with earnings and net present size.

Determine the gross present size for each friend.

Ed's Problem - Solution

Techniques

Geometric series
Binary search



Algorithm

For each friend, 'fill-up' tax bands one-by-one.

Start filling up the first tax band with any space left.

If the gift will not fit in this first band, work out the tax on this part of the gift and move onto the next tax band.

If the gift does fit, calculate tax, and tax on tax, etc. If that total would leave us in the same band, we are done.

If not, work out what portion of the tax will overlap, move to the next band and repeat.

F - Fine car dealership

Author: **Jim**

Overview

A map of a car dealership with doors, cars and walls.

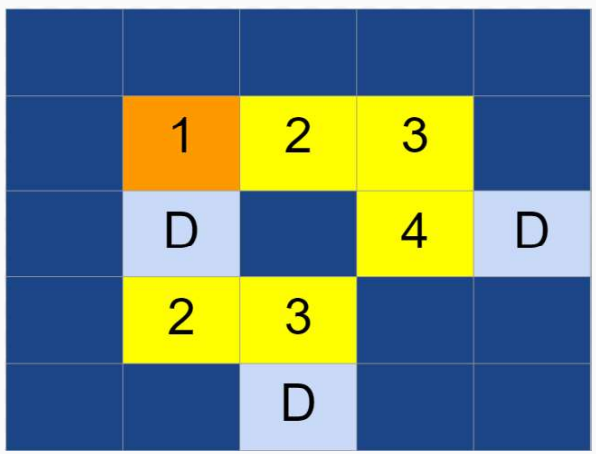
There can be many doors in the outer wall leading to the target.

Given the coordinates of a car in the dealership, how many cars must be moved in total.

Fine car dealership- Solution

Techniques

Dijkstra's algorithm
Breadth-first search



Algorithm

Read in the 'map' of the dealership and build a graph. Make a note of the doors on the edges.

Use Dijkstra's algorithm to find the distance to the target car.

Weight each node. 1 for a car, 0 for a door.

Push all of the edge doors onto a priority queue at once, distance 0

Starting a new search from each door is slow.

About 1,500 times slower, in fact.

See also: Sokoban for a harder challenge with the same idea

L - Lewis and the multi-level marketing

Author: **Robin**

Overview

N farmers each have a set, X

When asked, they will yield one item

But you can’t pick which one

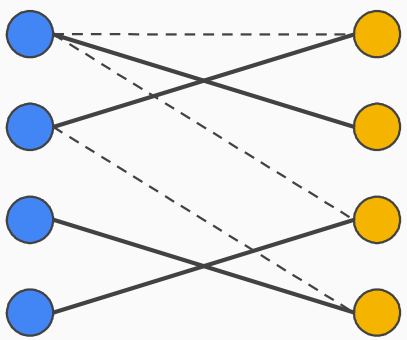
You want a certain number of each kind of bean

After utilising the farmers’ supplies, how many more beans will you need to barter for?

L - Solution

Techniques

Brute force
Combinations
Set cover



Algorithm

Each farmer will give the full amount of at least one kind of bean.

Proof by induction: either you already had enough, or getting another bean brings Lewis one step closer.

The worst case is when farmers collude:

Each picks a kind of bean to always give and puts it in set S

Cost = sum(beans \ S)

There are at most 2^B such sets---generate all of them, check if each makes a valid farmer selection, and take the smallest.

This is known as the set cover problem

Complexity: **$O(2^B \times N)$**