Sample title

Anonymous

Overleaf

2021

Aquarium Maze

Problem

- Input: grid of "." and "#" squares.
- Grid is filled with water from the top.
- Water can move down, left and right.

Solution

- Can simulate water by starting at a top square and then traversing the graph e.g. using BFS or DFS
  1. if square == "." and not yet visited
  2. visit all neighbours recursively
  3. add 1 to awnser

Gotchas

- Need to start once at every point on the top.
- Otherwise we might miss some air bubbles.

BicycleLock

Problem

- Input: Initial lock position I and final lock position F of length n.
- Move to final position by always turning two consecutive dials at once.

Solution

- Dial 1 can only be turned by turning dials 1 and 2.
- It needs to be turned from I_1 to I_2.
- After turning that, we have a new subproblem of length $n - 1$
- We can solve this recursively

Gotchas

- Always two ways to turn dials: Clockwise or Anti-clockwise.
- Need to check if dial n is at the right posititition in the end.

CompilersBrackets

Problem

- Check if given bracket pattern makes sense.
- {{}{} Does not make sense.
- {{}{{}}} Does make sense.

Solution

- Count number of currently open brackets.
- begin with open $= 0$
- "{" $\rightarrow$ open++
- "}" $\rightarrow$ open−−
- pattern invalid if open < 0 at any time.
- pattern invalid if open $! = 0$ in the end.

DamConstruction

Problem

- Given (n_1, n_2, n_4) lego bricks of size 1, 2 and 4.
- Build the highest wall of width w.

Solution

- Should always use bricks of higher size first to maintain flexibility.
- Greedy solution by using n_4 bricks, then filling up with n_2, then with n_1.

ExtravagantVoyage

Problem

- Given n items with happiness H and volume V
- Choose items with cumulative weight w
- Also called 0-1-Knapsack

Solution

- Recursive DP solution:
- Go through n items and start with remaining weight $r = w$.
- Recursively solve:
  - taking item: r -= W_i; $h += H_i$.
  - leaving item: r, h unchanged.
- Save states in dp-table.

Gotchas

- Not using a dp-table results in time limit exceeded.

FascinatingBooks

Problem

- Check if given list of strings contains every letter of the alphabet.

Solution

- Can concatenate strings and solve for a single string.
- For each char:
- If char is letter: Add lowercase version to set.
- Check if length of set is 26.

Gotchas

- Capitalization does not matter.
- Strings do not only contain letters

GoingHome

Problem

-

Solution

-

Gotchas

-

HiddenWords

Problem

- TODO: make description more clear.
- Given n strings, find a string that contains all n strings in the right order and where no character is part of 3 strings.

Solution

- Start with two first strings X and Y.
- If Y starts with X:
- Add X to solutionword, continue with rest of Y and next string.
- Else: Remove first letter of X and repeat process.

IntuitiveCitations

Problem

- Given a list of names. Print the lexicographically smallest surname and add " et al.".

Solution

- Remove part before space (prename).
- Find the lexicographically smallest string by sorting and taking the first element.
- print string + " et al.".

JollyFishing

Problem

-

Solution

-

Gotchas

-

KeyboardRobot

Problem

- Given a 6x6 keyboard layout of letters and some text.
- Find a way to move 2 fingers simultaneously such that the time is minimal to type the given text.

Insights

- Insight #1: the text is short, only 200 letters, so the maximum time is $(5 + 5) \cdot 200 = 2000$
- Insight #2: we can simulate it, but need fast way of prioritising interesting states

Solution

- Use a priority queue to track every "reasonable" reachable state
- A state is: (time, index in text, finger 1&2 target position and remaining movement)

LeaderboardPrediction

Problem

- Given the times you need to solve each of the n problems.
- Determine the minimal penalty you can get on the contest.

Solution

- The time you needed for the first problem will be added to the penalty of all problems you solve.
- It is always best to solve shortest problems first.
- Greedy solution: Sort problems by length, then simulate.