2018 Virginia Tech High School Contest Solution Outlines

The Judges

Dec 8, 2018

A – Climbing Worm – First solved at 0:01

Problem

Given a point that travels an upward distance a then a downward distance b in one iteration, output the number of iterations it takes for the point to reach a height h for the first time.

A – Climbing Worm – First solved at 0:01

Solution

- The problem can be solved with a loop: keep track of the current height and keep a counter for the number of times the point travels upwards.
- In each iteration of the loop, first add a to the current height and increment the counter. Then, check if the current height is greater than or equal to h, exiting the loop if so. Finally, subtract b from the current height.
- The answer is equal to the value of the counter upon exiting the loop, since h will be reached for the first time when the point is traveling upwards.
- Also possible using $\mathcal{O}(1)$ formula: $\max(\lceil \frac{h-a}{a-b} \rceil + 1, 1)$.

C – Math Homework – First solved at 0:04

Problem

Given three small positive integers b, c, d and a small non-negative integer l, output all non-negative solutions (x, y, z) to the diophantine equation $bx + cy + dz = l$

C – Math Homework – First solved at 0:04

Solution

- Worst case is $b = c = d = 1$, $l = 250$, which has 31 626 solutions.
- Corner case: $l = 0$
- Use 2 nested loops: outer loop iterates over x from $0 \ldots \lfloor \frac{l}{b} \rfloor$, inner loop iterates y from $0 \ldots \lfloor \frac{l-xb}{c} \rfloor$
- Check if $l - bx - cy$ is a multiple of d and output if so.

M – Competitive Arcade Basketball – First solved at 0:06

Problem

Given a list of up to $n = 100\,000$ names, given a list of up to $m = 200\,000$ pairs of names and points, tabulate the points for each name and report those names that achieved above a threshold p.

M – Competitive Arcade Basketball – First solved at 0:06

Solution

- The size of the input mandates that the tabulation happens in constant time, which requires the use of a data structure that allows you to increment a player's score in constant time, such as a hash map mapping player names to their cumulative score.
- Complexity $\mathcal{O}(n + m)$.
- Be careful to not output names more than once.

E – Broken Swords – First solved at 0:10

Problem

Given N 4-bit vectors given as 0/1 strings, you can exchange the bits in the first (top) and second (bottom) columns, and third (left) and fourth (right) columns, respectively, to maximize the number of all 0000 rows (full swords). Report that number, and report the number of remaining 0 below all 0000 rows in columns $1, 2$ and $3, 4$, respectively.

E – Broken Swords – First solved at 0:10

Solution

- Greedy.
- Sum the 0-bits in each column to obtain T, B, L, and R.
- Answer is $a = \min\left(\lfloor\frac{(T+B)}{2}\rfloor, \lfloor\frac{(L+R)}{2}\rfloor\right)$
- Remaining slats (0-bits) are $(T + B) - 2a$ and $(L + R) - 2a$, respectively.

K – Good Messages – First solved at 0:11

Problem

Given an offset o representing a rotation cipher, a message m, and a number of times that cipher is applied n, determine after each application of the cipher whether the number of vowels in the new message is at least as half the number of consonants. Then determine whether this holds for more than half of the steps.

K – Good Messages – First solved at 0:11

Solution

- Have one loop run n times with counters keeping track of the number of times the message was deemed 'good' against the total number of steps.
- At each iteration replace each character of the message with the one corresponding to the offset. This can be done in a variety of ways, one such way is to add the offset and mod the result so it wraps around.
- For each character replaced we check if it is a vowel or a consonant, and at the end of the message we check whether it is good or bad, updating our counters accordingly.
- Finally we output 'Boris' if the 'good' counter is greater than 'bad'

F – Bowser's Pipes – First solved at 0:13

Problem

Given a forest of trees, answer many queries of the following kind: For a given leaf u of one of the trees in the forest, find and output the leaf v that is part of the same tree, but that is closest to the root of the tree (breaking ties by assigned number).

F – Bowser's Pipes – First solved at 0:13

Solution

- To pass within time limit, must precompute: (a) which tree each node is in; (b) the leaf closest to the root of each tree.
- Use BFS (breadth-first) searches from the root of each tree, labeling each node with the tree it is in.
- Same BFS finds the leaf that is closest to the root and records it.
- Queries are answered with table lookups in constant time.
- Overall complexity is $\mathcal{O}(N)$.

G – Touchdown! – First solved at 0:14

Problem

Given a set of integers representing yards gained in American Football, determine the outcome of the drive.

G – Touchdown! – First solved at 0:14

Solution

- Accurately simulate play rules.
- Maintain current position p, number of plays q since last first down, yards s gained since last first down.
- Each play, increment current position p by number of yards gained y, increment s by y, increment number of plays q since last first down by 1.
- Then check conditions:
  - If more than 4 plays since last first down and less than 10 yards gained, output Nothing and stop.
  - If more at least 10 yards gained, reset q and s.
  - If position $p \geq 100$, output Touchdown and stop.
  - If position $p \leq 0$, output Safety and stop
- If nothing has been output so far, output Nothing and stop.

D – Pharmacy – First solved at 0:18

Problem

Simulate 2 task ("prescription") queues Q_s ("on-site") and Q_r ("remote") served by T servers ("technicians"), where Q_s takes priority over Q_r. Report average completion times for each queue for a workload of up to 100 000 tasks.

D – Pharmacy – First solved at 0:18

Solution

- Use discrete event simulation.
- DES is a technique that can simulate changes in the state of the physical world at discrete points in time by processing a series of events.
- DES involves an event queue (a time-sorted list of events), as well as data structure describing the state of the physical world as it is being simulated.
- Time "jumps" from event to event.

D – Pharmacy – First solved at 0:18

Solution

- Maintain sorted event queue with 3 event types:
  1. On-Site Prescription Arrival (t_s, d) if on-site prescription arrives at t_s.
  2. Remote Prescription Arrival (t_r, d) ditto.
  3. Prescription Filled (t_f, t_d, k) if a technician finishes at time t_f working on a prescription of type k that was dropped off at t_d.
- Sorting order of event types is by time (increasing), breaking ties by On-Site Arrival, Remote Arrival, and Filled in this order; breaking further ties by d.
- Maintain current on-site and remote prescription queues (FIFO).
- Maintain count of number of available technicians.
- Prime event queue with workload events, carefully breaking ties.

D – Pharmacy – First solved at 0:18

Solution

Simulate events as follows:

- If on-site/remote prescription arrives, check if technician is available.
  - If no technician is available, add prescription to respective prescription queue.
  - Else, decrement number of technicians and schedule Filled event.
- If a prescription is filled, update respective sums of completion times
- Then:
  - If on-site prescriptions are pending, schedule Filled event for first one.
  - Else, if remote prescriptions are pending, schedule Filled event for first one.
  - Else, increment number of technicians.

Output answer once event queue is exhausted.

I – The Grand Adventure – First solved at 0:22

Problem

- Given a string consisting of symbols that are from one of 3 types of different brackets (plus an optional, to-be-ignored delimiter), find out whether the string is a balanced bracket expression!
- Here, the bracket types are
Open	Close
$ (Money)	b Banker
* (Jewel)	j Jeweler
| (Incense)	t Trader
- Ignore symbols of type "Ground"

I – The Grand Adventure – First solved at 0:22

Solution

- Use a stack to keep track of backpack's content.
- Begin reading in line, ignoring delimiter.
- When encountering opening symbols (i.e. money), push them onto stack.
- When encountering a closing symbol (i.e. banker), pop the stack if its corresponding opening bracket peer is at the top of the stack, otherwise quit and output NO.
- If the end of the line is reached and the stack is empty, the sequence was balanced.

B – Pedal Power – First solved at 1:11

Problem

Given a multigraph with 2 types of weighed edges, “bike” and “nonbike”, find the shortest path on a given route subject to the constraint that the bicycle must be moved in order to use a “bike” edge.

B – Pedal Power – First solved at 1:11

Solution

- $n = 300$ allows precomputation of all-pairs shortest distance via Floyd-Warshall on the bike and non-bike graphs, respectively. (Dijkstra works as well.)
- Then, use dynamic programming to find the shortest time to move the bike from l_j to l_k for all location pairs (l_j, l_k) while moving from location a_i to a_{i+1} using edges on the non-bike/bike graphs.
- Complexity $\mathcal{O}(n^3 + n^2 z)$

H – Substitution Mania! – First solved at 1:47

Problem

Given an unordered set of substitution ciphers, find the unique ordering that encrypts a given plaintext to a given ciphertext. Use this ordering in reverse to decrypt a second ciphertext to the solution plaintext.

H – Substitution Mania! – First solved at 1:47

Solution - Insights

- Since a substitution cipher has only 26 inputs/outputs, only up to 26 unique plaintext/ciphertext character pairs need to be considered to find the sequence of ciphers, the rest of the characters in the sample plaintext/ciphertext may be discarded.
- Naive approach of trying out all possible orderings would try out all 12! permutations, which is too slow.

H – Substitution Mania! – First solved at 1:47

Solution

- Traverse the implicit graph created by the orderings of the ciphers in order to find the correct ordering.
- To pass within the time limit, must traverse the graph from both ends.

H – Substitution Mania! – First solved at 1:47

Solution

- To do the bidirectional search:
  - Find all intermediary texts R using $\lceil \frac{n}{2} \rceil$ decryptions from the ciphertext and all intermediary texts L using $\lfloor \frac{n}{2} \rfloor$ encryptions from the plaintext.
  - Find the intersection of these two sets $L \cap R$ which is guaranteed to contain 1 element. Combine encryptions and decryptions accordingly.
- Overall complexity is

$$\mathcal{O}\left(\left(\left\lceil \frac{n}{2} \right\rceil ! + \left\lfloor \frac{n}{2} \right\rfloor ! \right) \cdot \binom{n}{\lceil \frac{n}{2} \rceil} \right)$$

- Note: $\binom{12}{6} \cdot (6! + 6!) = 924 \cdot 1440 = 1\,330\,560$.

J – Holiday Stars – First solved at 2:27

Problem

Given a polygon whose center of mass moves along a parabola and which simultaneously rotates around its center of mass, find out which of the polygon's corner first exceeds a given x-coordinate and when.

J – Holiday Stars – First solved at 2:27

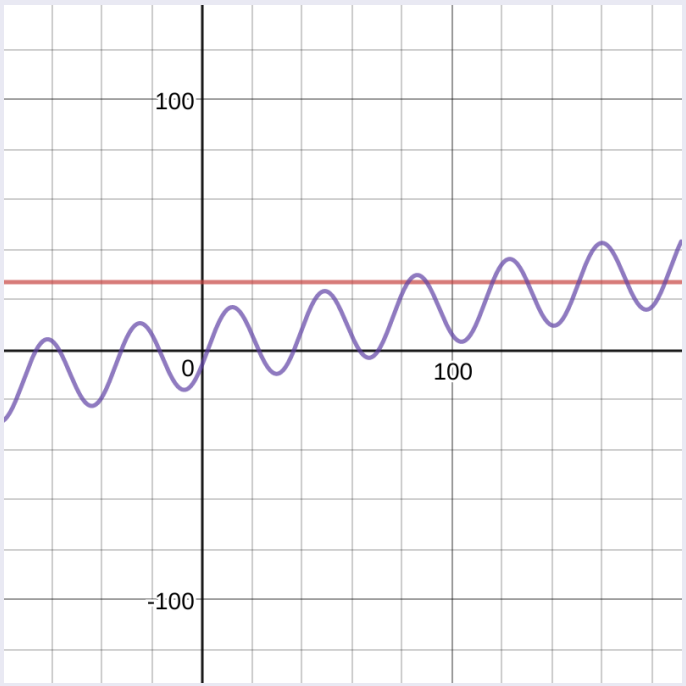
Solution

- Need only consider $x(t)$-coordinate at time t
- Horizontal velocity of center of mass is $v_x = v_0 \cos\theta$; convert to radians first.
- Location of each corner at time t is given as the sum of

$$X(t) = c_x + v_x t + d_x \cos(\omega t) + d_y \sin(\omega t)$$

where

  - c_x is the x-coordinate of the centroid.
  - (d_x, d_y) is the vector from the centroid to the corner.
  - Positive ω means clockwise (deviating from standard convention).



J – Holiday Stars – First solved at 2:27

Solution

- Apply trigonometric identity [1] to convert this linear combination to single sinusoid:

$$X(t) = c_x + v_x t + a \sin(\omega t + \phi)$$

  where $a = \sqrt{d_x^2 + d_y^2}$ and $\phi = \text{atan2}(d_x, d_y)$.
- Consider $X'(t) = v_x + a\omega \cos(\omega t + \phi)$. Since $|\cos(\omega t + \phi)| \leq 1$ it is true that $X(t)$ is monotonically increasing if $v_x \geq a\omega$. In this case, a simple binary search will find the answer.

J – Holiday Stars – First solved at 2:27

Solution

- Otherwise, the answer must lie in an ascending half period of the function, e.g., $T_0 \in [\frac{2\pi k-\phi}{\omega} - \frac{\pi}{\omega}, \frac{2\pi k-\phi}{\omega} + \frac{\pi}{\omega}]$ for some $k \geq \left\lfloor (\frac{w-a-c_x}{v_x}\omega + \phi)/(2\pi) \right\rfloor$ where the latter bound results from the line $c_x + v_x t + a$ which provides an upper bound for $X(t)$. Can use binary search within the first monotonic interval that crosses the wall, but discrete stepwise approach fits time limit and tolerance as well.

L – Bar Code – First solved at 4:24

Problem

Solve a 9×9 "Bar Code" puzzle which involves placing bars on a grid according to constraints.
See [URL]

L – Bar Code – First solved at 4:24

Solution

- Naive backtracking should time out.
- Need backtracking with some pruning.
- First try all ways of placing vertical bars in each row; prune if they create impossible situations for placing horizontal bars.
- Then try all ways of placing horizontal bars; no pruning necessary.
- Unoptimized Pypy 2 implementation runs in 8s.

References

[1] https://en.wikipedia.org/wiki/List_of_trigonometric_identities#Sine_and_cosine.