CTU Open 2021

Presentation of solutions

November 27, 2021

Cyanide Rivers

Cyanide Rivers

- Just count the largest consecutive block of zeroes...

Silver Star Stands Alone

Silver Star Stands Alone

- Implement brute-force.

Silver Star Stands Alone

- Implement brute-force.
- Use dynamic programming.

Silver Star Stands Alone

- Implement brute-force.
- Use dynamic programming.
- Alternative solution: Use time and precalculate answers.

Organ-free Man

Organ-free Man

- $$f(x) = \begin{cases} x! & \text{if } 0 \leq x \leq 9 \\ (x \bmod 10)! + f(\lfloor x/10 \rfloor) & \text{otherwise} \end{cases}$$

Organ-free Man

- $$f(x) = \begin{cases} x! & \text{if } 0 \leq x \leq 9 \\ (x \bmod 10)! + f(\lfloor x/10 \rfloor) & \text{otherwise} \end{cases}$$
- Iteration is not enough! $y = 10^9 = f$ (2233455555666677778888889999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999)

Organ-free Man

- Instead of adding digit i to the result for the $i+1$-th time, we add digit $i+1$ once, because $(i+1)i! = (i+1)!$

Organ-free Man

- Instead of adding digit i to the result for the $i+1$-th time, we add digit $i+1$ once, because $(i+1)i! = (i+1)!$
- $\Rightarrow$ Greedily add largest possible digits starting from the end

Organ-free Man

- Instead of adding digit i to the result for the $i+1$-th time, we add digit $i+1$ once, because $(i+1)i! = (i+1)!$
- $\Rightarrow$ Greedily add largest possible digits starting from the end
- Take care of $y = 1 = f(0)$

Terrace Hill

Terrace Hill

- Use a stack algorithm to find bridges.

Terrace Hill

- Use a stack algorithm to find bridges.

```
for(ll idx=0; idx<N; ++idx){
  ll height = a[idx];
    while(s.back().height < height)
       s.pop_back();
  if(s.back().height == height) {
    res += idx - s.back().idx - 1;
    s.back().idx = idx;
  } else {
    s.push_back({idx, height});
  }
}
```

Terrace Hill

- Use a stack algorithm to find bridges.

```
for(ll idx=0; idx<N; ++idx){
  ll height = a[idx];
    while(s.back().height < height)
       s.pop_back();
  if(s.back().height == height) {
    res += idx - s.back().idx - 1;
    s.back().idx = idx;
  } else {
    s.push_back({idx, height});
  }
}
```

- $O(N)$

Eidam Sand Lair

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;
    - if we go by foot,

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;
    - if we go by foot, we go by foot the whole way . . .

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;
    - if we go by foot, we go by foot the whole way ...
  - If the lift is above

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;
    - if we go by foot, we go by foot the whole way ...
  - If the lift is above
    - we go by foot if the lift is slower

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;
    - if we go by foot, we go by foot the whole way ...
  - If the lift is above
    - we go by foot if the lift is slower
    - possibly: we go by foot to the lift and use lift, if the lift is faster

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;
    - if we go by foot, we go by foot the whole way ...
  - If the lift is above
    - we go by foot if the lift is slower
    - possibly: we go by foot to the lift and use lift, if the lift is faster
    - if the lift is 'very' fast, then we call it and wait for it

Eidam Sand Lair

- How to get to the surface from floor Y_p underground while possibly using the lift?
- We can observe that only 3 solutions are reasonable:
  1. go only by foot,
  2. go only by lift,
  3. go by foot to lift and the rest by lift.
- Reasoning
  - If lift is below
    - if we use the lift, then it is not worth going any length by foot;
    - if we go by foot, we go by foot the whole way ...
  - If the lift is above
    - we go by foot if the lift is slower
    - possibly: we go by foot to the lift and use lift, if the lift is faster
    - if the lift is 'very' fast, then we call it and wait for it
- just compute all of these and take the minimum

Burizon Fort

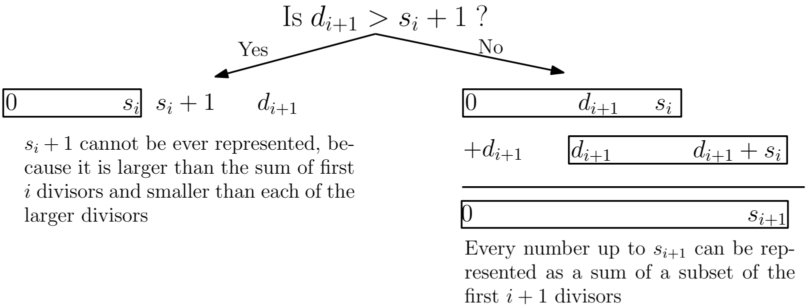
Burizon Fort

- Well-known: every $m \leq 10^{12}$ has at most 6 720 divisors.
- Divisors d_i sorted in ascending order.
- Let $s_i = d_1 + \cdots + d_i$.
- Claim: The number m is practical iff for the smallest i, such that $s_i >= m - 1$, every number from $\{1, \ldots, s_i\}$ can be represented as a sum of a subset of the first i divisors.

Burizon Fort

Proof of the Claim:

- We go through $i = 1, \ldots$ and notice that in the case when some number from $1, \ldots, s_{i+1}$ cannot be represented using the first $i+1$ divisors, then this number cannot be represented even if we allow the other divisors.



Screamers in the Storm

Screamers in the Storm

- **Task**: Count the number of sequences of integers of length n such that no two consecutives numbers have a common divisor greater than 1.
- The sequences can be long, up to 10^{18}, but contain only small numbers, up to 66.

Screamers in the Storm

- **Task**: Count the number of sequences of integers of length n such that no two consecutives numbers have a common divisor greater than 1.
- The sequences can be long, up to 10^{18}, but contain only small numbers, up to 66.

- **Solution**: Consider a graph, where integers up to 66 are vertices.
- We connect i and j by an edge if and only if $\gcd(i, j) = 1$.
- This means that any sequence can be created by walking along edges and it will always be valid.
- The number of sequences is the number of walks through this graph of length n.

Screamers in the Storm

- Let $M \in \mathbb{N}^{P \times P}$ be the incidence matrix of the graph for integers up to P.
  - Then $M_{i,j} = 1$ if and only if $\gcd(i, j) = 1$.
- The value of $M_{i,j}^{n-1}$ will be the number of walks of length n that start on i and end on j.
- Then sum of all elements of M^{n-1} is the number of valid sequences of length n.

Screamers in the Storm

- This can be shown be induction. $M^0 = E$ has zeroes everywhere, except ones on the diagonal.
- This corresponds to all sequences of length 1.
- Now if M^{n-1} is the number of sequences of length n, then M^n will be the number of sequences of length $n + 1$.
- $M^n_{i,j} = (M^{n-1} \cdot M)_{i,j} = \sum_{k=1}^{P} M^{n-1}_{i,k} \cdot M_{k,j}$ is the number of walks that start on i, then do $n - 1$ of steps ending on any k and then go directly from k to j.

Screamers in the Storm

- Use fast exponentiation to get M^n in $\mathcal{O}(\log(n))$ matrix multiplications.
- The complexity of the solution
  - Build the graph, that is compute $M_{i,j}$ using $\gcd(i,j)$ in $\mathcal{O}(P^2 \log(P))$ for all $1 \leq i,j \leq P$.
  - Then compute M^N in $\mathcal{O}(P^3 \log(N))$.
  - With $N \leq 10^{18}$ and $P \leq 66$ safely passes the time limit.

TomTom Cruise

TomTom Cruise

- **Task**: Find the cheapest path which can start and end on any vertex and must use at least one edge. The cost is the sum of costs of all edges + cost of the start and end.
- **Solution**: Be smart with Dijkstra.
- Several possible solutions.

TomTom Cruise

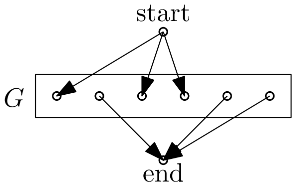
- Let G be the input graph. Use Dijkstra to search the state space where each vertex is a pair (s, v) where $s \in V(G)$ is the start of the path and $v \in V(G)$ is the current vertex of the path.
- Essentialy simulate exploring the graph using Dijkstra but keep track of which vertex was the starting one.

TomTom Cruise

- Let G be the input graph. Use Dijkstra to search the state space where each vertex is a pair (s, v) where $s \in V(G)$ is the start of the path and $v \in V(G)$ is the current vertex of the path.
- Essentialy simulate exploring the graph using Dijkstra but keep track of which vertex was the starting one.
- When v is first reached as (s_1, v) and (s_2, v) with $s_1 \neq s_2$, then the total cost of paths from (s_1, v) and (s_2, v) is the cost of the shortest walk that includes v and starts in s_1 and ends in s_2.
- It suffices to reach every v only twice using two different starts.
- The shortest valid path will be found when considering some v that belongs to it.
- The complexity is that of Dijkstra's algorithm: $\mathcal{O}((n + m)\log(n))$.

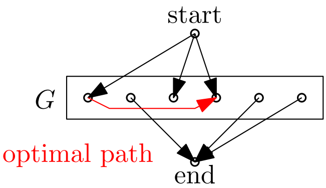
TomTom Cruise

- Time limit also allows slower solutions:
- Pick at random which vertices are possible starts and which possible ends.
- Each vertex is a possible start with probability $1/2$.
- Use Dijkstra to find the shortest path from all starts to any end.



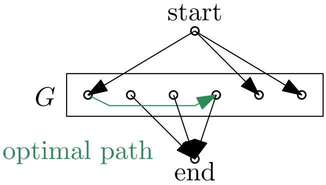
TomTom Cruise

- We missed the optimal path only if its both endpoints are both starts or both ends.
- This happens with probability $1/2$.



TomTom Cruise

- We missed the optimal path only if its both endpoints are both starts or both ends.
- This happens with probability $1/2$.



TomTom Cruise

- Repeating this k times gives the correct solution with probability 2^{-k}.
- With $k = 50$ we get the correct result in 99.9999999999999% and fit into the time limit.
- Total complexity is $\mathcal{O}(k(n+m)\log(n))$

TomTom Cruise

- We can use a similar approach without randomization
- Use any set of patterns which ensures that each pair of vertices is considered as start and end.

```
X.X.X.X.X.X.X.X.
XX..XX..XX..XX..
XXXX....XXXX....
XXXXXXXX........
```

- Correctness of this pattern is left as an excercise to the listener.
- Complexity: $\mathcal{O}((n+m)\log(n)^2)$.

Bread Pit

Bread Pit

- Observation: After meeting at least $\log(N)$ crossroads, there will be only 1 bread.

Bread Pit

- Observation: After meeting at least $\log(N)$ crossroads, there will be only 1 bread.
- Contract all paths of length 2 in tree: And simply simulate the process in $\mathcal{O} N \log(N)$

Tone Banks

Tone Banks

- Parsing of the old record – straightforward, but tedious implementation

Tone Banks

- Parsing of the old record – straightforward, but tedious implementation
- Output of the new record – recursively nest letters in a linear fashion

Tone Banks

- Parsing of the old record – straightforward, but tedious implementation
- Output of the new record – recursively nest letters in a linear fashion

"abbad":

```
###############
#.............#
#.#...........#
#.............#
#.###########.#
#.#.#########.#
#.###########.#
#.#.........#.#
#.#.#######.#.#
#.#.#.#.#.#.#.#
#.#.#######.#.#
#.#.#.....#.#.#
#.#.#######.#.#
#.#.........#.#
#.###########.#
#.............#
###############
```

Tone Banks

- Is the output grid large enough?

Tone Banks

- Is the output grid large enough?
- Maximal length of a word in the input grid $\lessapprox \frac{100\times 100}{4\times 4} = 625$

```
.....
.###.
.#.#.
.###.
.....
```

Tone Banks

- Is the output grid large enough?
- Maximal length of a word in the input grid $\lessapprox \frac{100 \times 100}{4 \times 4} = 625$

```
.....
.###.
.#.#.
.###.
.....
```

- The linear recursive approach needs 4 rows per letter in the worst case $\Rightarrow$ we can fit any word of length $\frac{3000}{4} = 750$

Win Diesel

Win Diesel

- Proceed “BFS-like” algorithm (possibly with priority queue)

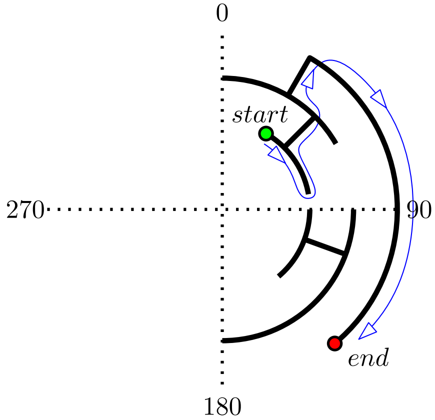
Win Diesel

- Proceed "BFS-like" algorithm (possibly with priority queue)
- Use LCA algorithm to find lengths of consecutively visited nodes.
- $ON \log(N)$

Mad Diamond

Mad Diamond

- Your task was to solve a brain teaser and figure out how to transport a marble from the start point to the end point. At the same time you need to minimize the total sum of degrees that the maze needs to be rotated by in order to achieve that.

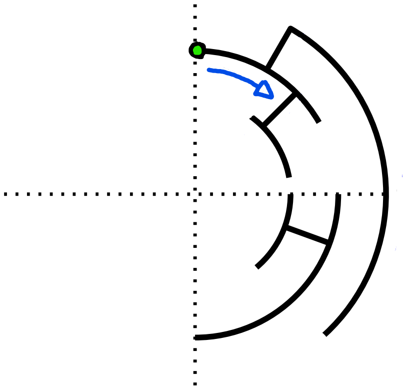


Mad Diamond

- The task can be modelled as a graph search.
- The state configuration consists of the layer at which ball is located, the rotation of the maze in degrees and the angular position of the ball in the maze, in degrees as well.
- There were at most 20 layers.
- Therefore as a result there might be at most $20 \cdot 360 \cdot 360 = 2592000$ reachable states.

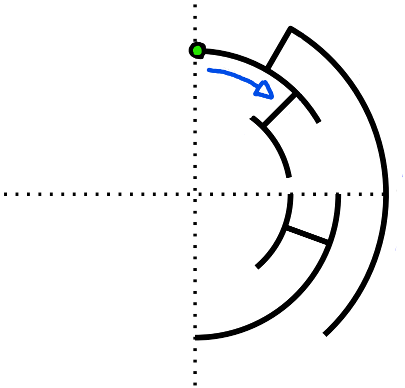
Mad Diamond

- There are a few tricky cases to handle. In the case below if the ball starts rolling down to the right we need to decide whether the ball should continue straight through the circular segment or rather fall through the radial segment.



Mad Diamond

- That needs to be decided based on the angle at which ball (considering the direction of gravity) meets the crossing between radial and circular segments.



Mad Diamond

- Another potential downfall is that at the start point the marble can start at such a position that the gravity will move it somewhere else right away.
- And similar thing applies to the finishing of the puzzle as well that the marble has to stop at the end point precisely, not just run through it which might make the puzzle impossible to solve.

Thank you for your attention!