Algolympics 2020

Solution Sketches

Kevin Atienza

Problem D: MTRCB

- Easiest problem
- Ignore the title. Don’t even read it!

Kevin Atienza

Problem D: MTRCB

- If else if else ...
- Check if ages are within threshold
- Respond accordingly

Kevin Atienza

Problem D: MTRCB

- Or use a data structure.
- Map each rating to its respective threshold

```
threshold = {'G': 0, 'PG': 13, 'R-13': 13, 'R-16': 16, 'R-18': 18}

if age < threshold[rating]:

    ...
```

Payton Yao

Problem K: I Brook the Code!

- 3 steps
  - Create pairs (height, width)
  - Sort
  - Print the sequence of widths

Payton Yao

Problem K: I Brook the Code!

- One line (after ignoring the first line of input)

```
print(*(w for h, w in sorted(zip(*[
        map(int, input().split())
        for i in range(2)
    ][::-1]))))
```

Kevin Atienza

Problem B: C.U.P.S.

- Any two cells can be flipped in two moves.

```
1 0 1 0 0 1 0
X X X X X 1 0
X X X X X 0 1
-------------
1 0 1 0 0 0 1
```

- Thus, for even no. of 0s, they can be flipped in at most n moves.

Kevin Atienza

Problem B: C.U.P.S.

- What about odd 0s?
- If m is even, then impossible. Every move flips even no. of cells, always.
- If m is odd, then just flip once, and the 0s become even!
  - Just make sure to prioritize flipping 0s.

Kevin Atienza

Problem B: C.U.P.S.

- Edge case: m = n
- Cannot do the two-move trick.
- Can only flip everything.

JD Dantes

Problem L: Break the Pattern!

- a, b, c are roots of (x - a)(x - b)(x - c) and any of its multiples.
- Conversely, every polynomial with roots a, b, c are multiples of (x - a)(x - b)(x - c).
  - Proof via polynomial division, and the fact that a polynomial with degree d has ≤ d roots.
- Gotcha: Remove duplicates to minimize degree!

Kevin Atienza

Problem A: The Slowden Files

- Determine if Levenshtein(s, t) is 0, 1, 2, 3 or more.
- Levenshtein(s, t) can be computed in O(|s||t|).
- Dynamic programming: Levenshtein(s’, t’) where s’ and t’ are prefixes.
- O(|s||t|) too slow!

Kevin Atienza

Problem A: The Slowden Files

- Instead, compute LevenshteinAtLeast(s', t', v) where v is up to 4.
- The key is, if ||s'| - |t'|| ≥ v, then the answer is automatically true.
- Hence, we may choose to only consider similar-length prefixes.
- O(|s| + |t|).

Kevin Atienza

Problem A: The Slowden Files

- In the real world, don’t implement this feature!
  - Passwords should not be stored in plaintext.

Barbara David

Problem E: A Floor of Many Doors

- Greedy: Open a door then close it immediately after passing.
- Except for k doors, which you can leave open.
- In your state, remember how many doors you've already opened
- Shortest path with weights up to 3 (or 4).
  - Can also use BFS with 4 (or 5) queues

Barbara David

Problem E: A Floor of Many Doors

- Tricky case! k = 1
- You can’t pass through double doors
- Solution:
  - Don’t pass through double doors.

Payton Yao, Kevin Atienza

Problem M: Thin Ice

- If either m or n is even, always possible.
- Find a cycle that passes through everything.
- From any starting position, just follow this cycle!

Payton Yao, Kevin Atienza

Problem M: Thin Ice

- If m and n are odd, then sometimes impossible.
- Checkerboard argument.

Payton Yao, Kevin Atienza

Problem M: Thin Ice

- If we start on black, impossible.
- So we must start at a white cell.

Payton Yao, Kevin Atienza

Problem M: Thin Ice

- Starting at white is always possible!
- Lots of cases.
  - Or maybe few, if you organize your thoughts better.

Payton Yao, Kevin Atienza

Problem M: Thin Ice

- Another edge case: r = 1 or c = 1.

Patrick Celon

Problem G: Generic Spy Movie

- Backtracking.
- enumerate_combinations(in_cast, out_cast, num).
- Recurses twice, similar to Pascal triangle

Josh Quinto, JD Dantes, Kevin Atienza

Problem J: A Cold Macchiato

- The i’th dispenser becomes the j’th dispenser with probability p[i] * q[i][j].
- There are 9 possibilities.
- In each one, we will have three temperatures t[k][1..3] and a corresponding probability p[k].

Josh Quinto, JD Dantes, Kevin Atienza

Problem J: A Cold Macchiato

- A strategy s[1..3] is such that s[1] + s[2] + s[3] = 1, s[i] ≥ 0.
  - Only proportion matters.
- The probability is the sum of all p[k] such that
  - $L \le s[1] * t[k][1] + s[2] * t[k][2] + s[3] * t[k][3] \le R$

Josh Quinto, JD Dantes, Kevin Atienza

Problem J: A Cold Macchiato

- Replacing s[3] = 1 - s[1] - s[2], we get a 2D system of half-planes. $ax + by \leq c$ or $ax + by \geq c$.
- The optimal solution is a point (x, y) somewhere.

Josh Quinto, JD Dantes, Kevin Atienza

Problem J: A Cold Macchiato

- Insight 1: The optimal solution lies in a convex region.

Josh Quinto, JD Dantes, Kevin Atienza

Problem J: A Cold Macchiato

- Insight 2: We can just take any vertex of the convex region as the candidate.

Josh Quinto, JD Dantes, Kevin Atienza

Problem J: A Cold Macchiato

- Solution:
  - Collect all half-planes.
  - Collect all intersections of pairs of half-plane edges.
  - Try all those candidates.
- The number of candidates is on the order $4*9^2/2$
  - $4*9^2/2 \simeq 200$
- O(1)

Josh Quinto

Problem C: Senpai

$P_i(0) = 0$ for all i from 1 to q $\quad \sum_{i=1}^{q} \left(\frac{d}{dt}P_i(t)\right)^2 \le g^2 \quad S_i(t) = F_i \cdot t + C_i$

- The best way to distribute your qualities *P* is to point it in the same direction as the vector *W*, and make the magnitude *gt*.
  - This is the best across all possible functions P
    - can be proven w/ basic analysis $||P(t)|| = ||W||gt$
  - Resulting effective quality score is
- Sum all components of S(t) (results in one linear function)

Josh Quinto

Problem C: Senpai

- Equation is now $||W||gt \geq \sum_{1}^{q} F_i(t) + \sum_{1}^{q} C_i$
- Solving t:

$$t = \frac{S_C}{g\,\|W\| - S_F}; S_F = \sum_{1}^{q} F_i; S_C = \sum_{1}^{q} C_i$$

- Problem occurs on very small $g\,\|W\| - S_F; S_F > 0$
- Triggers a *precision error* (even on double types)

Josh Quinto

Problem C: Senpai

- Issue: Numerically unstable
- Fix: Conjugate fraction to make denominator an integer

$$t = \frac{S_C*(g\|W\|+S_F)}{g^2\|W\|^2-S_F^2}; \|W\|^2 = \sum_1^q W_i^2$$

- Denominator is now an integer, so no precision loss.
- The sum above is now stable.
- BUT for negative S_F, this is now unstable.
  - Use the previous equation instead.

Josh Quinto

Problem C: Senpai

- Only a problem in C++
  - Python/Java has BigInteger/Decimal or something similar

JD Dantes

Problem F: One Great Grater

- Tricky cases
- Won’t spoil for now...
- Try it yourself first.
  - Try it on the mirror!
  - https://codeforces.com/group/fDKsZH3HKS/contests

Kevin Atienza

Problem I: Glory to Algotzka

- Given i, c, s, is there a subtree rooted at i with exactly c ‘C’ nodes and exactly s ‘S’ nodes?

Kevin Atienza

Problem I: Glory to Algotzka

- Crucial insight:
- For a given subtree size t (= c + s), let c_{min} and c_{max} be the minimum and maximum number of ‘C’ nodes in any subtree rooted at i of size t
- Then any c in $[c_{min}, c_{max}]$ is possible!

Kevin Atienza

Problem I: Glory to Algotzka

- Then any c in $[c_{min}, c_{max}]$ is possible!
- Proof:
  - Transform the min-tree into the max-tree one node at a time.
  - c will change from c_{min} to c_{max}.
  - Also, c only changes at most one each time.
  - Therefore, all c will be encountered!

Kevin Atienza

Problem I: Glory to Algotzka

- Compute $[c_{min}, c_{max}]$ for every pair (i, t) where $0 \le t \le \text{size}(i)$. Queries can now be answered in O(1).
- Computing c_{max} is equivalent to computing s_{min}, so we can just focus on c_{min} WLOG.

Kevin Atienza

Problem I: Glory to Algotzka

- $c_{min}(i, t)$ can be computed with DP, depending on the children nodes.
- E.g. for two children j and k:
- $c_{min}(i, t) = \min_u(c_{min}(j, u) + c_{min}(k, t - 1 - u)) + [i \text{ is a 'C'}]$.

Kevin Atienza

Problem I: Glory to Algotzka

- $c_{min}(i, t) = \min_u(c_{min}(j, u) + c_{min}(k, t - u)) + [i \text{ is a 'C'}]$.
- Naive analysis would show that this is $O(n^3)$
  - too slow
- But actually, if implemented well, it is $O(n^2)$!
- Overall complexity: $O(n^2 + q)$.

Kevin Atienza

Problem H: Maggie and Dana's Mass Supper

- Define f(i, j) to be the number of paths from (i, j) to the corner.
- Of course, f(i, j) = f(i, j-1) + f(i-1, j) (except on walls), but that’s too slow.

Kevin Atienza

Problem H: Maggie and Dana's Mass Supper

- Insight, only consider f(i, j) for the “corner cells”.
- There are only 2w such nodes.
- f(i, i) and f(i, l - w + i).

Kevin Atienza

Problem H: Maggie and Dana's Mass Supper

- Won’t spoil the rest for now...
- Try it yourself first.
  - Try it on the mirror!
  - https://codeforces.com/group/fDKsZH3HKS/contests

Thank you!

- **Kevin Charles Atienza**
- **Rene Josiah Quinto**
- **Joseph Daniel Dantes**
- **Marc Patrick Celon**
- **Payton Robin Yao**
- **Tim Joseph Dumol**
- **Barbara David**
- **Pio Fortuno III**
- **Samsung Testers**

- **A: The Slowden Files -** Atienza
- **B: C.U.P.S. -** Atienza
- **C: Senpai -** Quinto
- **D: MTRCB -** Atienza
- **E: A Floor of Many Doors -** David
- **F: One Great Grater -** Dantes
- **G: Generic Spy Movie -** Celon
- **H: Maggie and Dana's Mass Supper -** Atienza
- **I: Glory to Algotzka -** Atienza
- **J: A Cold Macchiato -** Quinto, Dantes, Atienza
- **K: I Brook the Code! -** Yao
- **L: Break the Pattern! -** Dantes
- **M: Thin Ice -** Yao, Atienza