PHS Computing Olympiad February 2025

Contest Editorial

A – Polly's New Laptop

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/A
Code:
Time Complexity: O(1)
Credits: William Park

Since the input command starts with the same five characters every time ("echo" and a space), simply output the input string with the first five characters removed.

B - Make the Grade

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/B
Code:
Time Complexity: O(n)
Credits: William Park

We only need to check if Polly can get an A or not. So, assume Polly gets a perfect 100% score on the final, and see if Polly ends up with an A. If Polly cannot get an A in the class even after a perfect final exam score, then no possible score on the final can give her an A.

C - Magical Mirror

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/C
Code:
Time Complexity: O(1)
Credits: William Park

The key observation here is that Polly can set her x-coordinate to any value she wants through a single reflection (note that Polly can choose any real number n for the reflection line). Similarly, Polly can set her y-coordinate to any value with only one reflection.

The only way Polly can save time is if her x-coordinate and/or y-coordinate is already the same as Calvin's. So, the answer is 2 seconds (one reflection per coordinate), minus 1 second if her x-coordinate is already correct, and minus 1 second if her y-coordinate is already correct.

D - Starr Showdown!

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/D
Code:
Time Complexity: O(a+c)
Credits: William Park

Note that the “supercharged” mechanic boils down to every third attack dealing double damage, regardless of misses. Since the order of misses does not matter, simply calculate the number of attacks it takes for Polly and Stanley to defeat each other, respectively. This can be done with modular arithmetic for an O(1) solution, but directly simulating the battle in O(a+c) will suffice due to the low constraints.

If Polly requires more attacks than Stanley, then she can never win, as she will be defeated first even if she does not miss any attacks. Otherwise, the answer is the difference between the two attack numbers.

E - Hexadecimal Hex

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/E
Code:
Time Complexity: $O(s_{len})$
Credits: William Park

Under this number system (also true for base-10 and base-16), the effective value of any digit will always exceed that of a digit to its right, so a greedy algorithm will suffice here. Use a dictionary/map (or other suitable structure) to keep track of which hexadecimal character corresponds to which decimal value. Starting from 15 and going from left-to-right, assign the highest possible number to each unique character in the string. Once done, calculate the final value of the hexadecimal number by multiplying the corresponding value of each digit by its place value.

F – Blocked Out

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/F
Code:
Time Complexity: $O(n^2)$
Credits: William Park

Going row-by-row, if a row contains an X but no Y, the Y must go in the blocked-out cell. Similarly, if a row contains a Y but no X, the X must go in the blocked-out cell. If both an X and Y are present in a row, there is nothing in the blocked-out cell. If neither an X nor Y are present in a row, this is the blocked-out row; save this one for last.

Use two arrays to keep track of which columns an X or Y have not appeared in. Once all rows except the missing row have been appropriately filled, there must be only one column where an X has not appeared in, and similarly only one column where a Y has not appeared in. Use this information to fill in the blocked-out row, completing the schedule.

G - Recursive Dartboard

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/G
Code:
Time Complexity: O(1)
Credits: William Park

Note that Calvin's recursive function simply calculates the minimum of $|x|$ and $|y|$. So, Calvin's score can be maximized by aiming at the "corners" of the dartboard - the intersection of the circle with the lines y = x and y = -x.

Since Calvin's function does not care about the signs of x & y, we can focus on the first quadrant and multiply our final answer by 4. The intersection of the circle with the line y = x in the first quadrant is at $(\sqrt{n/2}, \sqrt{n/2})$, and rounding down these coordinates gives us our first point (a, a) where $a = int(\sqrt{n/2})$. However, note that due to rounding, it is possible that $(a + 1, a)$ and $(a, a + 1)$ are also valid solutions that still lie within the circle; note that these still give the same final score of a. Test whether these points lie in the circle. If they do, then there are 3 * 4 = 12 solutions; otherwise, there are 1 * 4 = 4 solutions.

Note that n = 1 is a special case, since all five points in the circle (0,0), (0,1), (1,0), (0, -1), and (-1, 0) result in the same final score of 0. So, when n = 1, output "0 5" rather than using the above algorithm.

H1/2 - Bitstring Birthday Game

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/H1
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/H2
Code (H1):
Time Complexity (H1): O(a+b)
Code (H2):
Time Complexity (H2): O(a+b)
Credits: William Park

The key observation here is that AND operators will always take precedence over OR operators. So, a helpful way to visualize this problem is to imagine the OR gates as splitting the bitstring into groups. If one of these groups contains a 0, the entire group simplifies down to a 0, since anything AND 0 is 0. Conversely, the only way for a group to simplify down to a 1 is if every number in the group is a 1.

For Polly to succeed (H1), at least one of these groups must simplify to 1, since anything OR 1 is 1. If she has at least two OR gates, then as long as the bitstring contains a 1, she can surround the 1 with OR gates to succeed. If Polly has only one OR gate, then either the starting number or the ending number of the bitstring must be a 1, which will allow her to split it off into a group. If Polly has no OR gates, then every number in the bitstring must be a 1, as an AND gate will be placed between every number.

For Calvin to succeed (H2), all of the groups must simplify to 0. For this to be possible, every group must contain at least one 0. The number of groups is equal to $b + 1$, and if this number is greater than the number of zeros in the bitstring, then Calvin cannot succeed as he cannot place a zero in every group. Otherwise, it can be shown that it is possible for Calvin to create an arrangement of operators such that the final bitstring evaluates to 0. One such way to do so is to go through the bitstring from left-to-right and place an OR gate to the right of every zero. By the time the last OR gate is used, every group created so far will have at least one zero, and there will be at least one zero left in the bitstring for the final group (the last OR gate to the end of the bitstring).

I1/2 - High Rollers

Statement:
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/I1
https://codeforces.com/group/nxOH2ImmkR/contest/583932/problem/I2
Code (I1):
Time Complexity (I1): O(nlogn + n*q)
Code (I2):
Time Complexity (I2): O(nlogn + qlogn)
Credits: William Park

First, figure out the minimum number of turns Polly needs to defeat the enemy. Assuming Polly rolls the maximum possible dice value every time, the maximum possible damage she can deal per turn is the sum of her dice. Divide the health of the enemy by this number, rounding accordingly, to find the number of turns Polly needs.

To find the minimum number of dice, note that it is optimal for Polly to choose the highest-value dice to roll each time. So, sort the dice from greatest to lowest and loop through them, and keep a track of the number of dice and total dice values that Polly has rolled so far. If the die value times the minimum turns does not reach the enemy's health, move on to the next die; otherwise, use modular arithmetic to find the minimum number of times Polly needs to roll a specific die to deal enough damage.

This solution runs in O(nlogn) to sort the dice plus O(n) to loop through the dice for q queries, which results in O(nlogn + n*q) and is fast enough to pass I1. To optimize this solution for I2, calculate the prefix sum of the dice after sorting, which takes a negligible O(n) per test case. Then, perform binary search on the prefix sum to find the die that does not need to be rolled every turn, which can be done in O(logn) per query. This results in an overall O(nlogn + qlogn) time complexity, which is suitable for I2.

Credits

Thanks to the following people for making the first-every Poolesville High School Computing Olympiad possible!

Aanshi Patel - Problem Testing & Organization
Nicole Sabova - Problem Proofreading & Inspiration
Brooke Yin - Promo Banner Artist
Mr. Estep & Mr. Khetarpal - Creating the opportunity to run contests like this

Some cool stats: during the contest window, we had **20** participants who sent **255** code submissions and collectively solved a total of **79** problems!

Fun fact: The constraint on D for a & c was intended to be the maximum possible health of a brawler in Brawl Stars. Unfortunately this information is no longer accurate. Thanks Supercell.

Fun fact: I1 & I2 had to be redesigned two days before the competition because the problem originally didn't place a constraint on the number of turns. I realized much too late that you could just roll the highest-value dice every turn (and nothing else).

Unused concept art of Polly Programmer:

there are zero thoughts in her brain /affectionate