National Girls' Programming Contest 2022
(Online Preliminary)

Judge Notes

February 2023

1 Summary

The contest took place online on February 03, 2023. 243 teams participated in the contest. The problemset consists of 7 problems in total of varying difficulties. A short analysis of the problems are included in Section 2. Further statistics and the problems themselves can be found in the Contest Page.

2 Problemset Analysis

Implementations of the solutions described here can be found in the GitHub repository.

A. Python Interpreter

Author:	Rafid Bin Mostofa
Tester:	Abu Saleh
Alternate Writers:	Md. Rahat Islam, Shimul Sutradhar
Techniques:	String manipulation

Abridged Statement

Parse string X from a given line in the format `print("X")`

Solution

Considering L to be the length of the line, X will be the substring of the line starting from index 7 and ending at L-3 (0-indexed, inclusive). In Python, `line[7:-2]` should return the substring.

B. On My Way

Author:	Apurba Saha
Tester:	Tariq Bin Salam
Alternate Writers:	Rafid Bin Mostofa, Shimul Sutradhar
Techniques:	Dijkstra, DP

Abridged Statement

Count the number of paths from s to t in a connected weighted undirected graph such that, the path length differs from the shortest path length of s to t by at-most k.

Solution

First calculate the shortest path from s to every other vertex using Dijstra algorithm.

Let, $SP[i]$ be the shortest path length from s to i. As $k \leq 100$ and all weights are positive we can use DP.

Let, $dp[i][delta]$ be the number of paths from vertex i to t when current path length from s to $i = SP[i] + delta$.

From $dp[i][delta]$ the dp transition will be all the edges that have vertex i as one of its endpoint. Suppose there is an edge $i \rightarrow v$ with weight x. So we will go to the next state $dp[v][delta + SP[i] + x - SP[v]]$ if $delta + SP[i] + x - SP[v] \leq k$. Make sure to add 1 when you are in vertex t.

Time complexity: $O((n + m)log(n + m) + (n + m) * k)$.

C. Maximize Sum

Author:	Shimul Sutradhar
Tester:	Md. Rahat Islam
Alternate Writers:	Apurba Saha, Sabbir Rahman Abir
Techniques:	Dynamic Programming

Abridged Statement

Two integer arrays A and B given. You can add at most m element from B into A. Each element of B can be added at most once. You can remove at most k elements from array A. After performing the aforementioned operations output the maximum possible value of $\sum_{i=1}^{|A|} A_i \times i$.

Solution

Let F(i, j, x, y) = maximum possible value of $\sum_{i=1}^{|A|} A_i \times i$.

Here,

- i = index of array A.
- j = index of array B.
- x = Number of elements taken from array B and added to array A.
- y = Number of elements removed from array A.

At first sort array B in increasing order. Because if the value of $B[j]$ is maximized then the value of $i \times B[j]$ will be maximized.

For a new array A, index can have the following type of scenario.

- We can take the current i'th element from array A.

- We can remove i'th element if the value of y is less than equal to k.
- We can add j'th element of array B at the current possession of array A.
- We can skip current j'th element from B.

From those four scenarios, we have the following recurrence relation.

$$f(i,j,x,y) = max \begin{cases} 0, & \text{if i > n and j > m} \\ F(i+1,j,x,y) + A(i) \times (i+x-y), & \text{if i} \leq \text{n} \\ F(i+1,j,x,y+1), & \text{y < k} \\ F(i,j+1,x+1,y) + B(j) \times (i+x-y), & \text{j} \leq \text{m} \\ F(i,j+1,x+1,y) + B(j) \times (i+x-y), & \text{j} \leq \text{m} \end{cases}$$

By using those recurrence relations do dynamic programming to get the answer.

D. Greedy Grid Game

Author:	Abu Saleh
Tester:	Shimul Sutradhar
Alternate Writers:	Md. Rahat Islam, Mahdi Hasnat Siyam
Techniques:	Adhoc, Greedy

Abridged Statement

Given $n \times m$ grid, Aliban has to reach cell (n, m) from $(1, 1)$. Here she can go only right and down including only one diagonal move in her whole journey. Except for the diagonal moves, she cannot move left or up. Calculate the maximum score she can achieve.

Solution

Observe the moving condition of Aliban from the statement. There are limited movements available. All possible scenarios of movements-

1. For $n = 1$ or $m = 1$ only 1 path exists to reach the destination.
2. For $n = 2$ and $m = 2$ there are 4 possible paths.
3. For $n = 2$ or $m = 2$ there are 6 possible paths.
4. For $n > 2$ and $m > 2$ there are 8 possible paths.

Look into Figure 1 for a better understanding. Finally, iterate over all paths to calculate the score. From all scores, the maximum score will be the answer

E. GCD, Divisor, Count!

Author:	Md Sabbir Rahman
Tester:	Kazi Md Irshad
Alternate Writers:	Abu Saleh, Iftekhar Hakim
Techniques:	Number Theory, Combinatorics

Abridged Statement

Given integers N, A, B, count the number of N length positive integer arrays such that elements of the array divide B and the GCD of the elements is A.

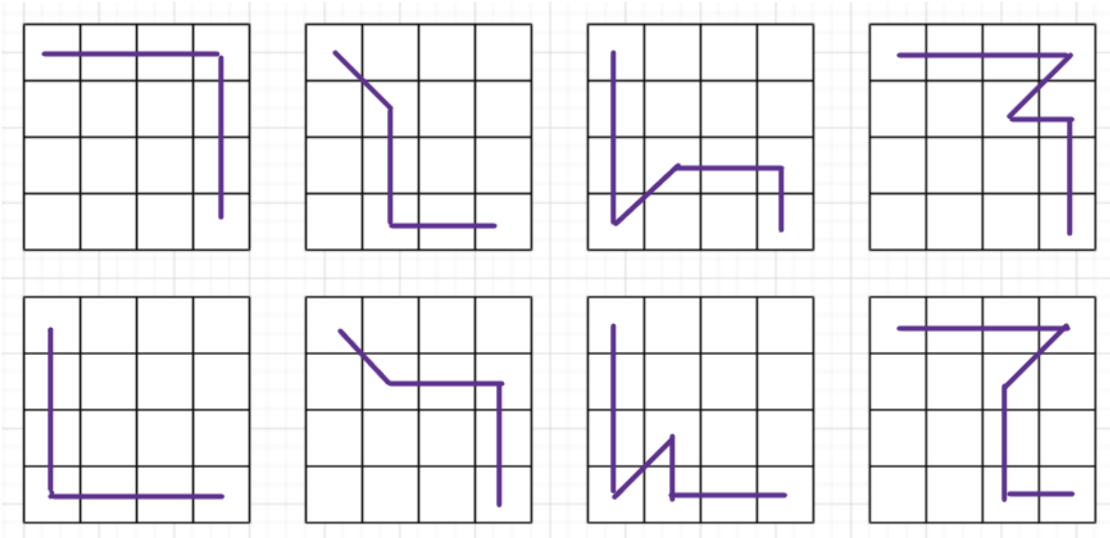

Figure 1: Possible paths in D. Greedy Grid Game

Solution

We first note that if A does not divide B, it is impossible to make required arrays. So in that case answer is 0. Otherwise, we know the entries of array are $A \cdot x_i$, in other words, we need to find count of n length arrays $[x_1, \cdots, x_n]$ such that x_i divides B/A and $gcd(x_1, \cdots, x_n) = 1$. There are two solutions to this, but for both, we'll need to factorize B/A to it's prime factors. We can use trial division upto $\sqrt{B/A}$ for that.

Note both cases require usage of modular exponentiation in logarithmic time, popularly known as bigmod.

Approach 1, using powers

Note that if $B/A = p_1^{e_1} \cdots p_k^{e_k}$, then we know each x_i can also be written as $x_i = p_1^{f_{i,1}} \cdots p_k^{f_{i,k}}$. Where $f_{i,j} \leq e_j$ and $\min(f_{1,j}, \cdots, f_{n,j}) = 0$. This is a counting problem now, we have to assign sequence $(f_{1,j}, \cdots, f_{n,j})$ satisfying those two conditions, for $j = 1, \cdots, k$.

If we relax the second condition, number of such sequence is $(e_j + 1)^n$. Among these there are some cases with no $f_{i,j} = 0$. Number of such case is e_j^n. Thus number of required sequences is $(e_j + 1)^n - e_j^n$.

We have to do this independently for each of $j = 1, \cdots, k$. Hence actual answer is $((e_1 + 1)^n - e_1^n) \cdots ((e_k + 1)^n - e_k^n)$

Approach 2, using inclusion-exclusion

Here we will first need to know that if $x = p_1^{e_1} \cdots p_k^{e_k}$ is the prime factorization of x then number of divisors of x is $(e_1 + 1) \cdots (e_k + 1)$. Let us call this $\tau(x)$ for short.

If we only try to find arrays $[x_1, \cdots, x_n]$ with elements dividing B/A. Then count of such array is $\tau(B/A)^n$. But we are overcounting cases where gcd is at least p_1 or at least p_2 or so on. To subtract these we subtract $\tau(B/(A \cdot p_1))^n + \cdots + \tau(B/(A \cdot p_k))$. But then we'll undercount those cases where gcd is at least $p_1 \cdot p_2$ and so on. This results in inclusion exclusion.

Ultimately the answer is to add $(-1)^m \cdot \tau(B/(A \cdot M))^n$ where M is a product of m size subset of distinct prime factors of B/A. To count this efficiently we'll have to iterate over all such M using bitmasks and use the formula of $\tau(x)$.

Complexity

if $\omega(x)$ = number of distinct prime factors of x, then
Time Complexity of first solution: $O(T \cdot (\sqrt{B/A} + \omega(B/A) \cdot \log N))$
Time Complexity of second solution: $O(T \cdot (\sqrt{B/A} + 2^{\omega(B/A)} \cdot (\omega(B/A) + \log N)))$

F. Cardboard Mountains

Author:	Md. Rahat Islam
Tester:	Rafid Bin Mostofa
Alternate Writers:	Abu Saleh, Kazi Md Irshad
Techniques:	Geometry

Abridged Statement

Given a few triangles on a XY-plane where the triangles touch base with the X axis and each triangle intersects only with the left and the right to it, compute the area the triangles cover together.

Solution

In Figure 2, the area of the purple polygon will be output. Here, these things should be observed,

1. A point p_i is considered left to p_j if $x_{p_i} < x_{p_j}$
2. A point p_i is considered right to p_j if $x_{p_i} > x_{p_j}$
3. There can be multiple coordinates with the same x-value. In this case, only considering the coordinate with the maximum y-value is to be considered as it will cover all the coordinates below while creating the shape.
4. Parts that overlap are to be added once.

At first, the coordinates are to be sorted based on their x-value. From the N coordinates, the coordinates with the highest y-values among the coordinates with the same x-values are to be taken. The intersecting points' coordinates of the adjacent would-be triangles' sides are found.

N triangles can be created. The area of each triangle can be found using,

$$\frac{1}{2}.height.base$$

Here *height* will be the y-value of the taken coordinate, and *base* will be the difference of x-values of the coordinates immediately left and right to the taken coordinate. The area of the overlapping triangles can be found in a similar manner, taking the y-value of the intersecting point as the *height* and the difference of x-values of the coordinates immediately left and right to it as the *base*.

The final area would be,

$$A - B$$

where,
A =Summation of the areas of the triangles made from the given coordinates.

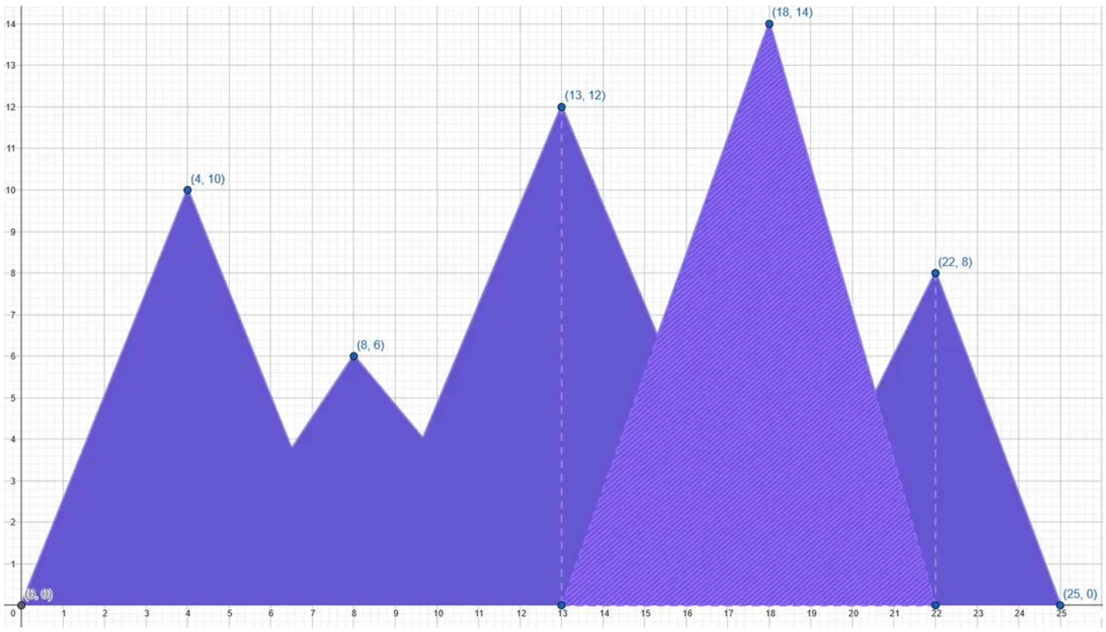

Figure 2: Cardboard Mountains

B =Summation of the areas of the triangles made from the intersection point's coordinates.

Another way of solving this problem would be to list the intersection points with the filtered given points and intersection points along with $(0, 0)$ and $(X, 0)$. Then sort all the points based on their x-values. The final area will be the area of the polygon made using the sorted coordinates in a clockwise or anticlockwise manner. The area can be easily found using the shoelace method.

G. Blanket

Author:	Kazi Md Irshad
Tester:	Mahdi Hasnat
Alternate Writers:	Rafid Bin Mostofa
Techniques:	Math

Abridged Statement

Given a cubic object which is foldable by length or width keeping volume same, compute the minimum number of folds necessary for it to fit in another cubic object.

Solution

The suitcase can be re-oriented for the blanket to fit inside. There are $3! = 6$ possible orientations of the suitcase. Try all of the permutations of (X, Y, Z). Now the problem becomes: given two triplets (X, Y, Z) and (L, W, H) find minimum number of operations to make $L \leq X$, $W \leq Y$ and $H \leq Z$ where operations are $(L := \frac{L}{2}, H := H \times 2)$ or $(W := \frac{W}{2}, H = H \times 2)$.

Let k_1 be the largest non-negative integer such that $H \cdot 2^{k_1} \leq Z$. Similarly let k_2 and k_3 be the smallest non-negative integers such that $\frac{L}{2^{k_2}} \leq X$ and $\frac{W}{2^{k_3}} \leq Y$.

If $H > Z$ or $k_1 < k_2 + k_3$ then it is impossible to fit the blanket. Otherwise we can fit it in $k_2 + k_3$ folds.

The complexity of the solution is $O(\log_2(\max(X, Y, Z)))$ per test case.