Osijek Competitive Programming Camp Winter 2023
Day 9: Magical Story of LaLa
Solutions

Problem & Solution Author: Aeren

February 26, 2023

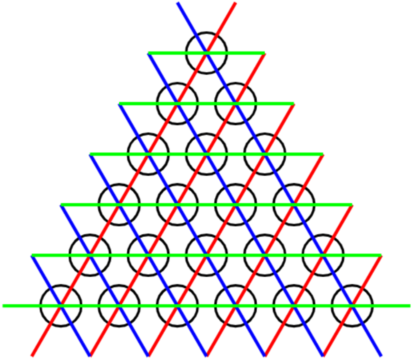
(C) LaLa and Lamp

- First Solved By: **NewTrend** at 00:28.

(C) LaLa and Lamp

Problem Description

You're given a binary triangular grid. Determine whether you can make it all 0 by flipping one of $3N$ row arbitrary many times.



(C) LaLa and Lamp

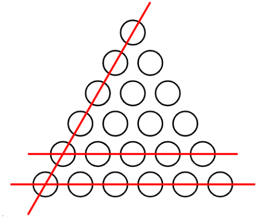
- The operations commute: the order of operations doesn't matter.

(C) LaLa and Lamp

- The operations commute: the order of operations doesn't matter.
- Two of the same operation cancels each other.

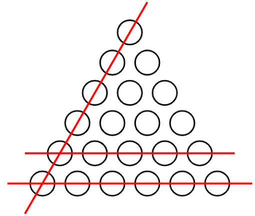
(C) LaLa and Lamp

- The operations commute: the order of operations doesn't matter.
- Two of the same operation cancels each other.
- Fixing two longest rows in one direction and the longest row in another direction uniquely determines the rest.



(C) LaLa and Lamp

- The operations commute: the order of operations doesn't matter.
- Two of the same operation cancels each other.
- Fixing two longest rows in one direction and the longest row in another direction uniquely determines the rest.



Time Complexity: $O(N^2)$

(E) LaLa and Monster Hunting (Part 1)

- First Solved By: **HoMaMaOvO** at 00:38.

(E) LaLa and Monster Hunting (Part 1)

Problem Description

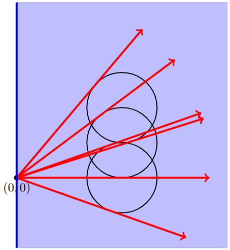
You're given N circles. Determine if their convex hull contains the origin.

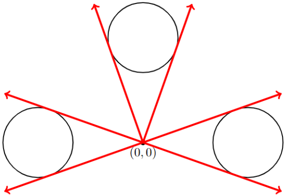
(E) LaLa and Monster Hunting (Part 1)

- If any circles contain the origin in its interior or on its boundary, the answer is "YES".

(E) LaLa and Monster Hunting (Part 1)

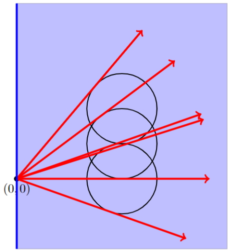
- If any circles contain the origin in its interior or on its boundary, the answer is "YES".
- Otherwise, each circles have 2 tangent rays starting from the origin. The answer is "YES" if and only if no half-plane contains all $2N$ rays.

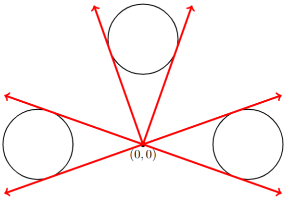




(E) LaLa and Monster Hunting (Part 1)

- If any circles contain the origin in its interior or on its boundary, the answer is "YES".
- Otherwise, each circles have 2 tangent rays starting from the origin. The answer is "YES" if and only if no half-plane contains all $2N$ rays.

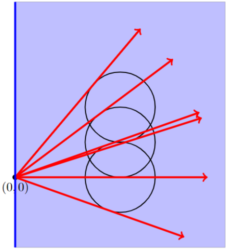


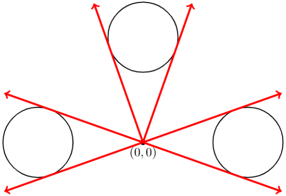


- It's safe to use doubles in computation due to the distance condition.

(E) LaLa and Monster Hunting (Part 1)

- If any circles contain the origin in its interior or on its boundary, the answer is "YES".
- Otherwise, each circles have 2 tangent rays starting from the origin. The answer is "YES" if and only if no half-plane contains all $2N$ rays.





- It's safe to use doubles in computation due to the distance condition.

Time Complexity: $O(N \cdot \log N)$ from sorting rays.

(E) LaLa and Monster Hunting (Part 1)

Alternative Solution

There exists an $O(N \cdot \log N)$ algorithm computing the convex hull of N circles.

Reference paper: "A convex hull algorithm for discs, and applications" by David Rappaport

Construct the hull, then check that for each boundary segment and arc, directed counter-clockwise, the origin lies on the left.

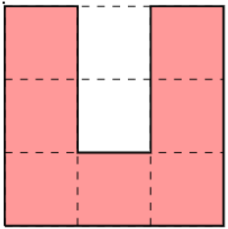
(D) LaLa and Magic Stone

- First Solved By: **HoMaMaOvO** at 01:14

(D) LaLa and Magic Stone

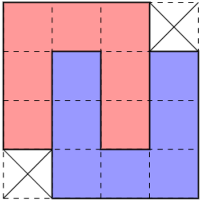
Problem Description

Given cells in a rectangular grid where some cells are unavailable, find the number of ways to partition the available cells into U-shaped pieces.



(D) LaLa and Magic Stone

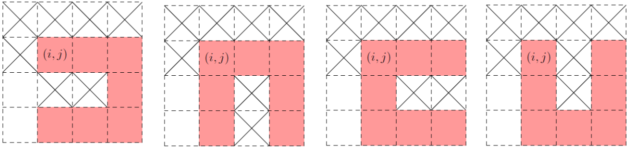
- The key observation is that merging two intertwined pieces produces a unique partitioning strategy, if there is one. We'll call this a **merged piece**.



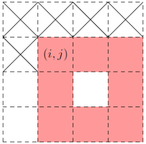
- The proof of the above statement naturally yields the algorithm to solve this problem.

(D) LaLa and Magic Stone

- Suppose that there exists a valid partitioning and an available cell. Let (i, j) be the lexicographically smallest available cell.
- If there is exactly one way to cover (i, j), cover it and go back to step 1.

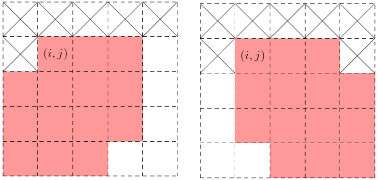


- Otherwise, the following 8 cells must be available.



(D) LaLa and Magic Stone

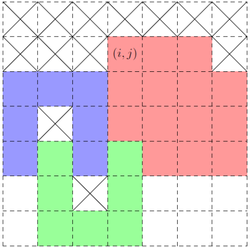
- Now there are two cases.
- Case 1: $(i+1, j+1)$ is available.
- The piece covering (i, j) has to be the merged piece. There are two possible ways that the merged piece cover (i, j). Call them configuration A (the left one) and B (the right one) respectively.



- When either of the configuration is invalid, the case becomes trivial. Now assume that both are valid.

(D) LaLa and Magic Stone

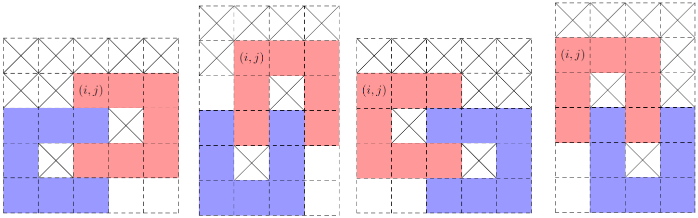
- If we can't fit a piece in the 3 by 3 square (with its top left corner) at $(i+1, j-3)$, we're forced to choose configuration B, and we go back to the start.
- Now assume we can. Then the only way to cover $(i, j+3)$ with configuration B is the following, and it is easy to verify that configuration A doesn't work on it. Therefore, we choose A or B depending on the availability of the following partitioning.



- (End of case 1)

(D) LaLa and Magic Stone

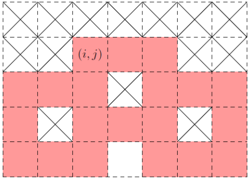
- Case 2: $(i+1, j+1)$ is unavailable.
- There are 4 ways to cover all cells in the 3 by 3 square (with its top left corner) at (i, j). Call them configuration L, BL, R, BR, respectively.



- If exactly one of the configuration is valid, use it, then go back to the start.

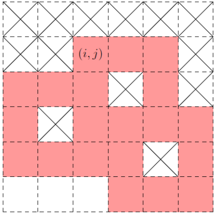
(D) LaLa and Magic Stone

- Now assume that at least 2 of them are valid. Since L and BL cannot be both valid, and the same for R and BR, we only have to consider 4 cases.
- Suppose L and R are valid, and assume that L is chosen. Then it's impossible to cover both $(i+1, j+3)$ and $(i+2, j+4)$ at once, so this case is impossible. Similar argument works when R is chosen instead. Therefore, this case is impossible.



(D) LaLa and Magic Stone

- Suppose L and BR are valid, and assume that L is chosen.

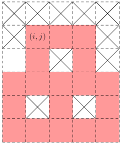


Then it's impossible to cover both $(i+1, j+3)$ and $(i+2, j+4)$, so this case is impossible. Similarly, assuming R is chosen, $(i+1, j-1)$ and $(i+2, j-2)$ cannot both be covered as well, so this case is impossible.

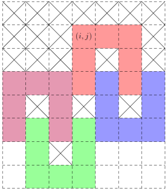
- The case when R and BL are valid is impossible as well by a similar argument.

(D) LaLa and Magic Stone

- Finally, suppose BL and BR are valid.



It's easy to see that if $(i+1, j-1)$ is available, there are no valid partitioning. Now, using a similar argument to the A & B case earlier, we chose BL or BR depending on the availability of the following partitioning:



(G) LaLa and Divination Magic

- First Solved By: **NewTrend** at 02:01

(G) LaLa and Divination Magic

Problem Description

Given a set of solutions for a 2-SAT formula, recover the formula, or report that there isn't one.

(G) LaLa and Divination Magic

- A literal in a 2-SAT formula is either a variable or a negation of a variable. Each literal L has an associated variable $V(L)$ and an associated value $E(L)$, either true or false depending on whether it's a positive literal or a negative literal.

(G) LaLa and Divination Magic

- A literal in a 2-SAT formula is either a variable or a negation of a variable. Each literal L has an associated variable $V(L)$ and an associated value $E(L)$, either true or false depending on whether it's a positive literal or a negative literal.
- A literal L is said to **imply** another literal M if for each solution where $V(L)$ is set to $E(L)$, $V(M)$ is set to $E(M)$.

(G) LaLa and Divination Magic

- A literal in a 2-SAT formula is either a variable or a negation of a variable. Each literal L has an associated variable $V(L)$ and an associated value $E(L)$, either true or false depending on whether it's a positive literal or a negative literal.
- A literal L is said to **imply** another literal M if for each solution where $V(L)$ is set to $E(L)$, $V(M)$ is set to $E(M)$.
- Let F be the 2-SAT formula where we've added every clause of form $\neg L \vee M$.

(G) LaLa and Divination Magic

- A literal in a 2-SAT formula is either a variable or a negation of a variable. Each literal L has an associated variable $V(L)$ and an associated value $E(L)$, either true or false depending on whether it's a positive literal or a negative literal.
- A literal L is said to **imply** another literal M if for each solution where $V(L)$ is set to $E(L)$, $V(M)$ is set to $E(M)$.
- Let F be the 2-SAT formula where we've added every clause of form $\neg L \vee M$.
- If there is a corresponding 2-SAT formula, we've included all of its clauses. The extra clauses added won't produce any contradiction either.

(G) LaLa and Divination Magic

- A literal in a 2-SAT formula is either a variable or a negation of a variable. Each literal L has an associated variable $V(L)$ and an associated value $E(L)$, either true or false depending on whether it's a positive literal or a negative literal.
- A literal L is said to **imply** another literal M if for each solution where $V(L)$ is set to $E(L)$, $V(M)$ is set to $E(M)$.
- Let F be the 2-SAT formula where we've added every clause of form $\neg L \vee M$.
- If there is a corresponding 2-SAT formula, we've included all of its clauses. The extra clauses added won't produce any contradiction either.
- Thus, the 2-SAT formula exists if and only if the set of solutions of F is equal to the input.

Enumerating all solutions of a 2-SAT formula

(G) LaLa and Divination Magic

Enumerating all solutions of a 2-SAT formula

1. Let G be the directed graph whose vertices are the literals and for each clause $L \vee M$, there are two edges $\neg L \to M$ and $\neg M \to L$.

(G) LaLa and Divination Magic

Enumerating all solutions of a 2-SAT formula

1. Let G be the directed graph whose vertices are the literals and for each clause $L \vee M$, there are two edges $\neg L \to M$ and $\neg M \to L$.
2. If there exists a literal L such that L and $\neg L$ lies in the same SCC, there are no solution.

(G) LaLa and Divination Magic

Enumerating all solutions of a 2-SAT formula

1. Let G be the directed graph whose vertices are the literals and for each clause $L \vee M$, there are two edges $\neg L \to M$ and $\neg M \to L$.
2. If there exists a literal L such that L and $\neg L$ lies in the same SCC, there are no solution.
3. Every assignment where there are no path from true to false is valid.

(G) LaLa and Divination Magic

Enumerating all solutions of a 2-SAT formula

1. Let G be the directed graph whose vertices are the literals and for each clause $L \vee M$, there are two edges $\neg L \to M$ and $\neg M \to L$.
2. If there exists a literal L such that L and $\neg L$ lies in the same SCC, there are no solution.
3. Every assignment where there are no path from true to false is valid.
4. Sweep through each literal L in topological order, and if it's unassigned, set it to true and false, and recurse on each case.

(G) LaLa and Divination Magic

Enumerating all solutions of a 2-SAT formula

1. Let G be the directed graph whose vertices are the literals and for each clause $L \vee M$, there are two edges $\neg L \to M$ and $\neg M \to L$.
2. If there exists a literal L such that L and $\neg L$ lies in the same SCC, there are no solution.
3. Every assignment where there are no path from true to false is valid.
4. Sweep through each literal L in topological order, and if it's unassigned, set it to true and false, and recurse on each case.
5. If it's set to true, the literals reachable from it must also be set to true.

(G) LaLa and Divination Magic

Enumerating all solutions of a 2-SAT formula

1. Let G be the directed graph whose vertices are the literals and for each clause $L \vee M$, there are two edges $\neg L \to M$ and $\neg M \to L$.
2. If there exists a literal L such that L and $\neg L$ lies in the same SCC, there are no solution.
3. Every assignment where there are no path from true to false is valid.
4. Sweep through each literal L in topological order, and if it's unassigned, set it to true and false, and recurse on each case.
5. If it's set to true, the literals reachable from it must also be set to true.

Time Complexity: $O(N \cdot M^2/w)$

(F) LaLa and Monster Hunting (Part 2)

- First Solved By: **Three Konjaks** at 03:28

(F) LaLa and Monster Hunting (Part 2)

Problem Description

Given a simple graph, count the number of subgraphs isomorphic to the graph in the statement.

(F) LaLa and Monster Hunting (Part 2)

The model solution sequentially computes the following values.

1. For each vertex u, the number of 3-cycles passing through u.
2. For each undirected edge e, the number of 3-cycles passing through e.
3. For each directed edge $e = (u, v)$, the number of subgraphs with 4 vertices $0, 1, 2, 3$ and 5 edges $(0, 1), (1, 2), (2, 3), (3, 0), (1, 3)$ such that u and v corresponds to 0 and 1 respectively.
4. For each directed edge $e = (u, v)$, the number of subgraphs with 5 vertices $0, 1, 2, 3, 4$ and 6 edges $(0, 1), (1, 2), (2, 3), (3, 0), (1, 4), (4, 2)$ such that u and v corresponds to 0 and 1 respectively.
5. For each vertex u, the number of answer with tail length $1, 2, 3$ such that u lies at the end of the tail.

Time Complexity: $O(n + m \cdot \sqrt{m})$

(I) LaLa and Spirit Summoning

- First Solved By: **HoMaMaOvO** at 02:41

(I) LaLa and Spirit Summoning

Problem Description

Given an edge-colored graph, find the minimum degree of freedom of a graph whose edges all have distinct color, that can be obtained by deleting some edges from the original graph.

(I) LaLa and Spirit Summoning

- Each edges either contribute to -1 degree of freedom or 0, and those -1 edges form a matroid. (Rigidity matroid)

(I) LaLa and Spirit Summoning

- Each edges either contribute to -1 degree of freedom or 0, and those -1 edges form a matroid. (Rigidity matroid)
- A set of edges is independent IFF every subgraph S satisfies $2 \cdot |V| - 3 \geq |E|$. (Laman's theorem)

(I) LaLa and Spirit Summoning

- Each edges either contribute to -1 degree of freedom or 0, and those -1 edges form a matroid. (Rigidity matroid)
- A set of edges is independent IFF every subgraph S satisfies $2 \cdot |V| - 3 \geq |E|$. (Laman's theorem)
- Independence oracle can be built by checking if 2-to-1 matching between vertices and edges still exists after quadrupling the new edge. (Pebble game algorithm)

(I) LaLa and Spirit Summoning

- Each edges either contribute to -1 degree of freedom or 0, and those -1 edges form a matroid. (Rigidity matroid)
- A set of edges is independent IFF every subgraph S satisfies $2 \cdot |V| - 3 \geq |E|$. (Laman's theorem)
- Independence oracle can be built by checking if 2-to-1 matching between vertices and edges still exists after quadrupling the new edge. (Pebble game algorithm)
- Matroid intersection algorithm + rigidity oracle with the pebble game algorithm gives a solution of complexity $O(N^2 \cdot M)$.

(H) LaLa and Harvesting

- First Solved By: **Zagreb** at 04:51

(H) LaLa and Harvesting

Problem Description

Given a vertex-weighted graph which is a union of

1. a cactus,
2. a cycle passing through all leaves of the DFS-tree of the cactus, and
3. a tree whose non-leaf vertices have degree ≥ 12,

find an independent set with the maximum sum of weight.

(H) LaLa and Harvesting

- If there were no third stage, and the graph on the first phase were a tree, this graph is known as a **halin graph** and has treewidth ≤ 3.

(H) LaLa and Harvesting

- If there were no third stage, and the graph on the first phase were a tree, this graph is known as a **halin graph** and has treewidth ≤ 3.
- If there were no third stage, for each bag in the halin graph decomposition, you insert the DFS-tree root of the relevant cycles. The resulting decomposition has treewidth ≤ 4.

(H) LaLa and Harvesting

- If there were no third stage, and the graph on the first phase were a tree, this graph is known as a **halin graph** and has treewidth ≤ 3.
- If there were no third stage, for each bag in the halin graph decomposition, you insert the DFS-tree root of the relevant cycles. The resulting decomposition has treewidth ≤ 4.
- Merging a graph with treewidth $\leq A$ and a graph with vertex cover $\leq B$ result in a graph with treewidth $\leq A + B$.

(H) LaLa and Harvesting

- If there were no third stage, and the graph on the first phase were a tree, this graph is known as a **halin graph** and has treewidth ≤ 3.
- If there were no third stage, for each bag in the halin graph decomposition, you insert the DFS-tree root of the relevant cycles. The resulting decomposition has treewidth ≤ 4.
- Merging a graph with treewidth $\leq A$ and a graph with vertex cover $\leq B$ result in a graph with treewidth $\leq A + B$.
- Since the tree on the third phase has vertex cover of size $\max(1, (k-1)/11)$, we can construct the decomposition of the input graph with width ≤ 13. Now you can find the answer in $O(N \cdot 2^{5+(k-1)/11} \cdot (5 + (k-1)/11))$ time.

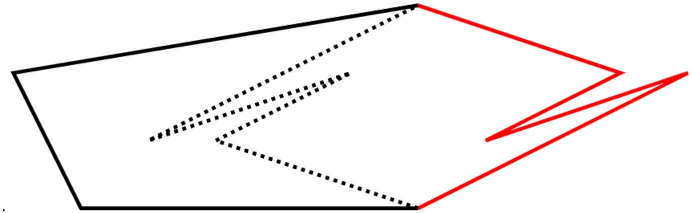
(A) LaLa and Magic Circle (LiLi Version)

- First Solved By: –

(A) LaLa and Magic Circle (LiLi Version)

Problem Description

Construct a polygon with $\leq 1,000$ vertices which has a sequence of the flip operations of length between $120,000$ and $1,000,000$.



(A) LaLa and Magic Circle (LiLi Version)

Here's one possible construction.

(A) LaLa and Magic Circle (LiLi Version)

Here's one possible construction.

- Let N be a positive integer divisible by 4 and d_i be the direction vector of the i-th edge in CCW.

(A) LaLa and Magic Circle (LiLi Version)

Here's one possible construction.

- Let N be a positive integer divisible by 4 and d_i be the direction vector of the i-th edge in CCW.
- (Group 1) For each $0 \leq i \leq N/2 - 2$, $d_i = (i+1, (i+1)^2)$.

(A) LaLa and Magic Circle (LiLi Version)

Here's one possible construction.

- Let N be a positive integer divisible by 4 and d_i be the direction vector of the i-th edge in CCW.
- (Group 1) For each $0 \leq i \leq N/2 - 2$, $d_i = (i+1, (i+1)^2)$.
- (Group 2) For each $N/2 - 1 \leq i \leq N - 2$ such that i is odd, $d_i = (1, 0)$.
- (Group 3) For each $N/2 - 1 \leq i \leq N - 2$ such that i is even, $d_i = (N/2, (N/2)^2)$.

(A) LaLa and Magic Circle (LiLi Version)

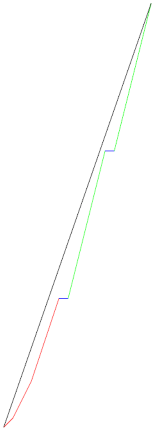
Here's one possible construction.

- Let N be a positive integer divisible by 4 and d_i be the direction vector of the i-th edge in CCW.
- (Group 1) For each $0 \leq i \leq N/2 - 2$, $d_i = (i+1, (i+1)^2)$.
- (Group 2) For each $N/2 - 1 \leq i \leq N - 2$ such that i is odd, $d_i = (1, 0)$.
- (Group 3) For each $N/2 - 1 \leq i \leq N - 2$ such that i is even, $d_i = (N/2, (N/2)^2)$.

Sliding down the first edge in the group 2 through all $N/2 - 1$ in group 1, and then sliding down the remaining $N/4 - 1$ edges in the group 2 one by one in order gives a sequence of operations of length $N^2/8 - 1$, which is equal to $12\,499$ for $N = 1\,000$.

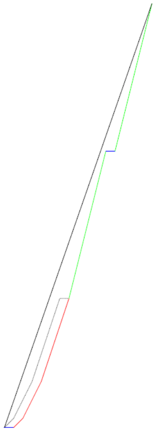
(A) LaLa and Magic Circle (LiLi Version)

[Group 1] [Group 2] [Group 3]



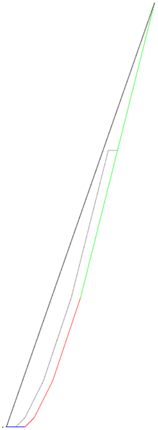
$\rightarrow$

3 Operations



$\rightarrow$

4 Operations



(A) LaLa and Magic Circle (LiLi Version)

The construction credit goes to "Polygons Needing Many Flipturns" by Therese Biedl.

(J) LaLa and Magical Beast Summoning

- First Solved By: –

(J) LaLa and Magical Beast Summoning

Problem Description

Solve the range query problem of the non-commutative and non-associative binary operation $\mathrm{Combine}$.

(J) LaLa and Magical Beast Summoning

Assume every cell lies in the summoning field $\mathcal{F}(M, E, V)$.

- We define
  - $-\mathcal{C}(L, A, I) = \mathcal{C}(L, I, A)$,
  - $\mathcal{C}(L_0, A_0, I_0) + \mathcal{C}(L_1, A_1, I_1) = \mathrm{Combine}(\mathcal{C}(L_0, A_0, I_0), -\mathcal{C}(L_1, A_1, I_1))$, and
  - $e = \mathcal{C}(0, 3, -3)$, which is valid.

(J) LaLa and Magical Beast Summoning

Assume every cell lies in the summoning field $\mathcal{F}(M, E, V)$.

- We define
  - $-\mathcal{C}(L, A, I) = \mathcal{C}(L, I, A)$,
  - $\mathcal{C}(L_0, A_0, I_0) + \mathcal{C}(L_1, A_1, I_1) = \mathrm{Combine}(\mathcal{C}(L_0, A_0, I_0), -\mathcal{C}(L_1, A_1, I_1))$, and
  - $e = \mathcal{C}(0, 3, -3)$, which is valid.
- With some algebra, it can be proved that
  1. $k_0 * \mathcal{C}(L_0, A_0, I_0) + k_1 * \mathcal{C}(L_1, A_1, I_1) = k_2 * (\mathcal{C}(L_0, A_0, I_0) + \mathcal{C}(L_1, A_1, I_1))$ for some integer $0 < k_2 < M$,
  2. $\mathcal{C}(L_0, A_0, I_0) + (\mathcal{C}(L_1, A_1, I_1) + \mathcal{C}(L_2, A_2, I_2)) = k * ((\mathcal{C}(L_0, A_0, I_0) + \mathcal{C}(L_1, A_1, I_1)) + \mathcal{C}(L_2, A_2, I_2))$ for some integer $0 < k < M$,
  3. $e + \mathcal{C}(L, A, I) = k_0 * (\mathcal{C}(L, A, I) + e) = k_1 * \mathcal{C}(L, A, I)$ for some integer $0 < k_0, k_1 < M$, and
  4. $\mathcal{C}(L, A, I) + (-\mathcal{C}(L, A, I)) = k * e$ for some integer $0 < k < M$.

(J) LaLa and Magical Beast Summoning

- Note that the density of $\mathcal{C}(L, A, I)$ is the same as $k * \mathcal{C}(L, A, I)$ for all integer $0 < k < M$.

(J) LaLa and Magical Beast Summoning

- Note that the density of $\mathcal{C}(L, A, l)$ is the same as $k * \mathcal{C}(L, A, l)$ for all integer $0 < k < M$.
- Given l and r, our goal is to find the density of the cell

$$(\cdots((C_l - C_{l+1}) - C_{l+2}) - \cdots) - C_{r-1} = k * (C_l - (C_{l+1} + \cdots + C_{r-1}))$$

for some integer $0 < k < M$, which is equal to the density of

$$C_l - (C_{l+1} + \cdots + C_{r-1})$$

(J) LaLa and Magical Beast Summoning

- Note that the density of $\mathcal{C}(L, A, I)$ is the same as $k * \mathcal{C}(L, A, I)$ for all integer $0 < k < M$.
- Given l and r, our goal is to find the density of the cell

$$(\cdots((C_l - C_{l+1}) - C_{l+2}) - \cdots) - C_{r-1} = k * (C_l - (C_{l+1} + \cdots + C_{r-1}))$$

for some integer $0 < k < M$, which is equal to the density of

$$C_l - (C_{l+1} + \cdots + C_{r-1})$$

- The second property ensures that querying in segment tree will find the result of $C_{l+1} + \cdots + C_{r-1}$ times some non-zero constant, and the first property will ensures that subtracting it from C_l will find the desired result times some non-zero constant.

(J) LaLa and Magical Beast Summoning

Time Complexity: $N + Q \cdot \log N$ from building and querying the segment tree.

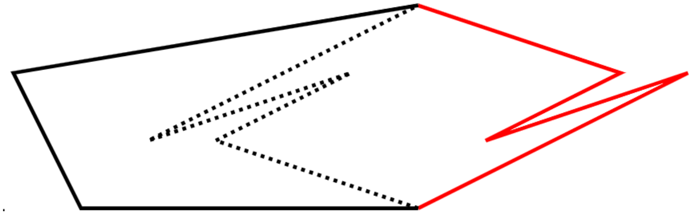
(B) LaLa and Magic Circle (LaLa Version)

- First Solved By: –

(B) LaLa and Magic Circle (LaLa Version)

Problem Description

Given a polygon, find the exact shape and location of the convex polygon obtained by applying flip operations.

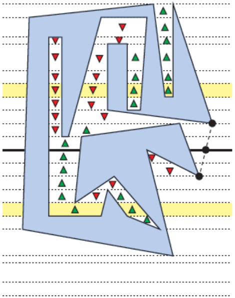


(B) LaLa and Magic Circle (LaLa Version)

- Proving the claims in the statement yields the algorithm to solve this problem.

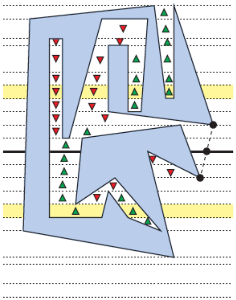
(B) LaLa and Magic Circle (LaLa Version)

- Proving the claims in the statement yields the algorithm to solve this problem.
- The final "shape" is fixed since the area always increases and it never changes the multiset of slopes of directed edges.



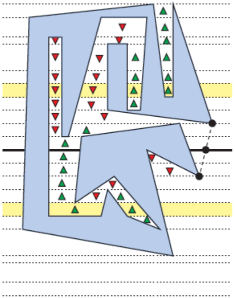
- Consider the horizontal trapzoidal decomposition of the outer-region of the polygon.

(B) LaLa and Magic Circle (LaLa Version)



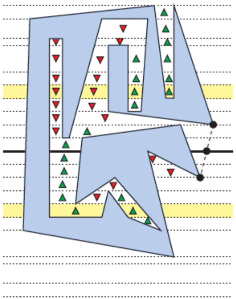
- Consider the horizontal trapzoidal decomposition of the outer-region of the polygon.
- For each finite regions, label it as up or down, depending on the direction you have to go to reach any infinite region.

(B) LaLa and Magic Circle (LaLa Version)



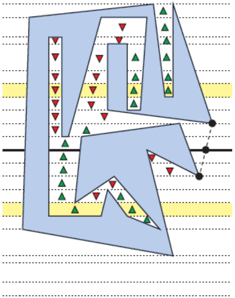
- Consider the horizontal trapzoidal decomposition of the outer-region of the polygon.
- For each finite regions, label it as up or down, depending on the direction you have to go to reach any infinite region.
- Let U be the sum of the heights of up-regions, and Y be the maximum y-coordinate. Then $U + Y$ is invariant throughout the operations.

(B) LaLa and Magic Circle (LaLa Version)



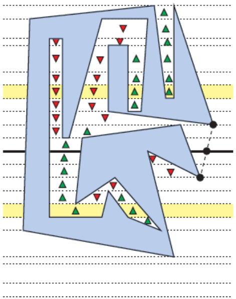
- Consider the horizontal trapzoidal decomposition of the outer-region of the polygon.
- For each finite regions, label it as up or down, depending on the direction you have to go to reach any infinite region.
- Let U be the sum of the heights of up-regions, and Y be the maximum y-coordinate. Then $U + Y$ is invariant throughout the operations.
- As $U = 0$ for the final convex polygon, this sum immediate gives the final maximum y-coordinate.

(B) LaLa and Magic Circle (LaLa Version)



- Consider the horizontal trapzoidal decomposition of the outer-region of the polygon.
- For each finite regions, label it as up or down, depending on the direction you have to go to reach any infinite region.
- Let U be the sum of the heights of up-regions, and Y be the maximum y-coordinate. Then $U + Y$ is invariant throughout the operations.
- As $U = 0$ for the final convex polygon, this sum immediate gives the final maximum y-coordinate.
- Repeat the same process with vertical trapzoidal decomposition to compute the maximum x-coordinate.

(B) LaLa and Magic Circle (LaLa Version)



- Consider the horizontal trapzoidal decomposition of the outer-region of the polygon.
- For each finite regions, label it as up or down, depending on the direction you have to go to reach any infinite region.
- Let U be the sum of the heights of up-regions, and Y be the maximum y-coordinate. Then $U + Y$ is invariant throughout the operations.
- As $U = 0$ for the final convex polygon, this sum immediate gives the final maximum y-coordinate.
- Repeat the same process with vertical trapzoidal decomposition to compute the maximum x-coordinate.

Reference Paper: Flipturning Polygons

Thanks for participating!