OCPC

Day 6: Dilhan Salgado Contest

Universal Cup Stage 5: Osijek

Analysis

I

Given a permutation, you may use this operation any # of times:

- Pick a nonempty prefix. Reverse it and the corresponding suffix.

How many permutations can you generate this way?

a b c d | e f

↓

d c b a | f e

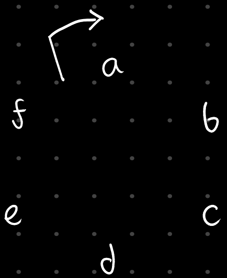
$\boxed{\text{I}}$

Given a permutation, you may use this operation any # of times:

- Pick a nonempty prefix. Reverse it and the corresponding suffix.

How many permutations can you generate this way?

Arrange the permutation on a circle



a b c d e f

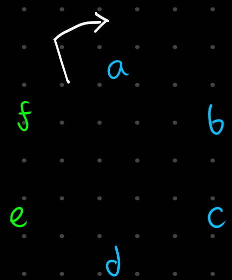
$\boxed{\text{I}}$

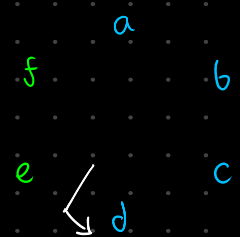
Given a permutation, you may use this operation any # of times:

- Pick a nonempty prefix. Reverse it and the corresponding suffix.

How many permutations can you generate this way?

Operating means choosing a different starting point & reversing direction





a b c d | e f

d c b a | f e

$\boxed{\text{I}}$

Given a permutation, you may use this operation any # of times:

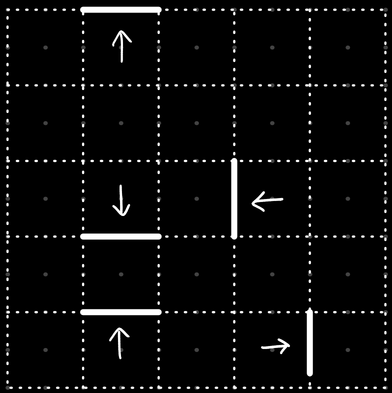
- Pick a nonempty prefix. Reverse it and the corresponding suffix.

How many permutations can you generate this way?

$n > 2 \rightarrow$ Can choose any starting point & direction.

G

Given a $n \times n$ grid of square-shaped countries. Each country chooses at most one of its edges and puts up a wall. No two walls touch. What is the max # of walls?



G

Given a $n \times n$ grid of square-shaped countries. Each country chooses at most one of its edges and puts up a wall. No two walls touch. What is the max # of walls?

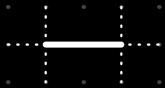
Forget about ownership for now.

How to get an upper bound?

G

Given a $n \times n$ grid of square-shaped countries. Each country chooses at most one of its edges and puts up a wall. No two walls touch. What is the max # of walls?

Each corner can be in at most one wall



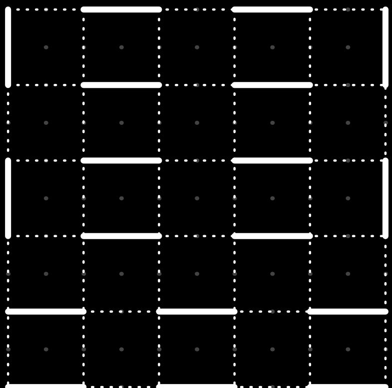
$\rightarrow$ answer $\leq$ (# of corners)/2

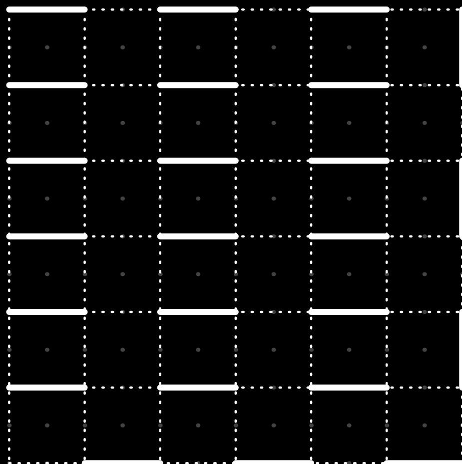
(the problem is exactly bipartite matching btw)

G

Given a $n \times n$ grid of square-shaped countries. Each country chooses at most one of its edges and puts up a wall. No two walls touch. What is the max # of walls?

$\left\lfloor (\text{\# of corners})/2 \right\rfloor$ walls

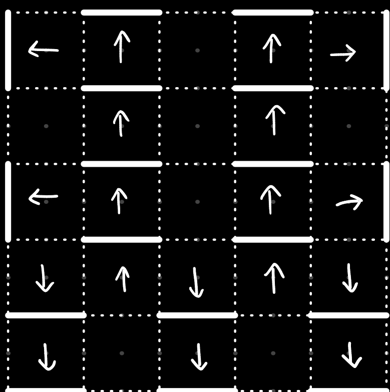


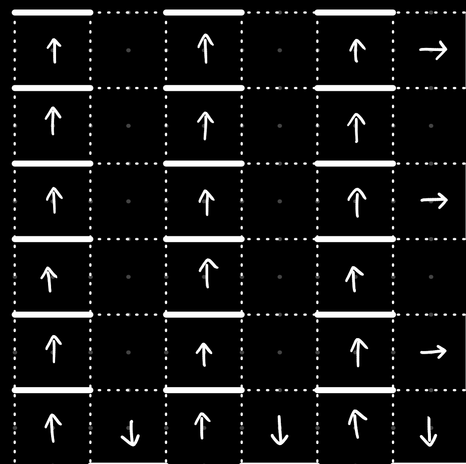


Achievable for both even, odd.

$\boxed{G}$ Given a $n \times n$ grid of square-shaped countries. Each country chooses at most one of its edges and puts up a wall. No two walls touch. What is the max # of walls?

Ownership works out too:





Achievable for both even, odd.

$\boxed{F}$

Given a string s, count the # of palindromic substrings. with length 5

$|s| \leq 10^6$

alphabet size ≈ 100

$\boxed{F}$ Given a string s, count the # of palindromic substrings.

$|s| \leq 10^6$
alphabet size ≈ 100

Must avoid $|s| \cdot 100^2$ time complexity

Must avoid $|s| \cdot 100$ space complexity

$\boxed{F}$ Given a string s, count the # of palindromic substrings.

$|s| \leq 10^6$
alphabet size ≈ 100

Sweep left-to-right, maintain:

- $dp_1[a]$ – # of subsequences "a" up to current point
- $dp_2[a][b]$ – # of subsequences "ab" up to current point
- $dp_3[a][b]$ – —— " —— "ab?" —— " ——
- $dp_4[a][b]$ – —— " —— "ab?b" —— " ——

$\boxed{F}$ Given a string s, count the # of palindromic substrings.

$|s| \leq 10^6$
alphabet size ≈ 100

Sweep left-to-right, maintain:

$dp_1[a]$ - # of subsequences "a" up to current point

$dp_2[a][b]$ - # of subsequences "ab" up to current point

$dp_3[a][b]$ - —— " —— "ab?" —— " ——

$dp_4[a][b]$ - —— " —— "ab?b" —— " ——

$dp_1[a]$, $dp_2[a][b]$: c – current character

$dp_2[a][c] \mathrel{+}= dp_1[a] \ \forall a$, $dp_1[c] \mathrel{+}= 1$

$\boxed{F}$ Given a string s, count the # of palindromic substrings.

$|s| \leq 10^6$

alphabet size ≈ 100

We update $dp_3[a][b]$ (# of "ab?") at the next b.

How many "ab?"-s have appeared since the last b?

$$dp_3[a][b] \mathrel{+}= dp_2[a][b] \cdot |i - last_b| \qquad \forall a$$

Calculate before updating $dp_2[a][b]$

b ________ b

$dp_3[a][b]$

$\boxed{F}$ Given a string s, count the # of palindromic substrings.

$|s| \leq 10^6$

alphabet size ≈ 100

We update $dp_3[a][b]$ (# of "ab?") at the next b.

How many "ab?"-s have appeared since the last b?

$$dp_3[a][b] \mathrel{+}= dp_2[a][b] \cdot |i - last_b| \qquad \forall a$$

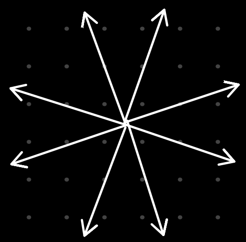
Calculate before updating $dp_2[a][b]$

dp_4, answer updated similarly

Be sure to update them in the correct order

J

Visit every row and every column of a $n \times n$ board with a piece that moves as follows:

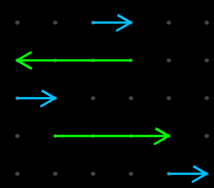


$\boxed{J}$ Visit every row and every column of a $n \times n$ board with a piece that moves as described above.

Look at only x-coordinate.

Can jump $-3, -1, +1, +3$

Must visit every square exactly once.



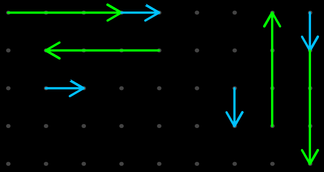
$\boxed{J}$ Visit every row and every column of a $n \times n$ board with a piece that moves as described above.

y must make a long jump when x makes a short jump (& vice versa)

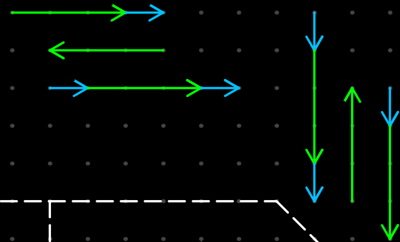
J

Visit every row and every column of a $n \times n$ board with a piece that moves as described above.

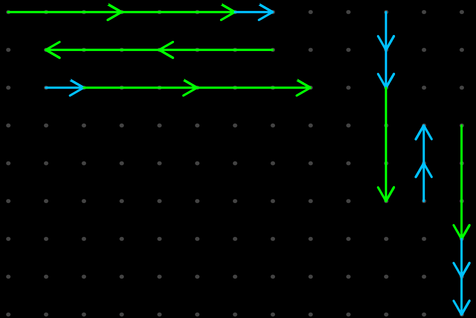
$n = 5$



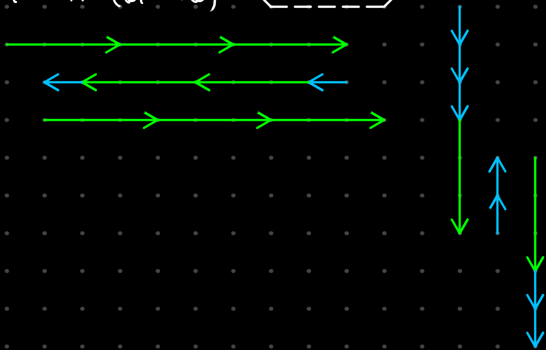
$n = 7$ (or 6)



$n = 9$ (or 8)



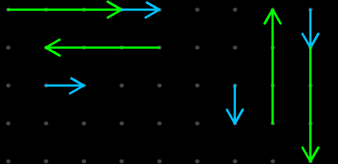
$n = 11$ (or 10)



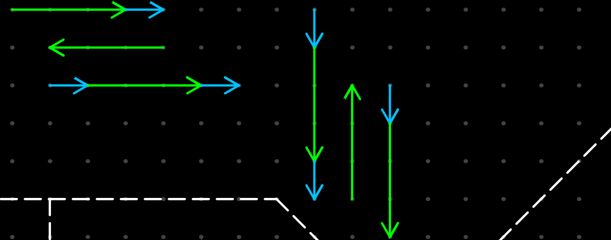
$\boxed{J}$

Visit every row and every column of a $n \times n$ board with a piece that moves as described above.

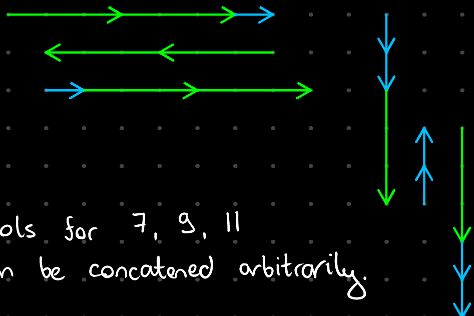
$n = 5$



$n = 7$ (or 6)

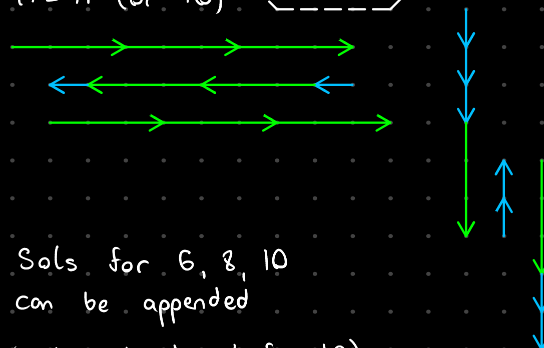


$n = 9$ (or 8)



Sols for 7, 9, 11 can be concatened arbitrarily.

$n = 11$ (or 10)



Sols for 6, 8, 10 can be appended

(Not pictured: sol for 12)

A

Given a tree, every vertex has a value $< 2^{25}$. Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\quad n \leq 10^5$

$\boxed{A}$

Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\quad n \leq 10^5$

Try separating the bits.

$\boxed{A}$

Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\qquad n \leq 10^5$

$\sum A_{uv}^2$

If i-th and j-th bit are set in A_{uv},

it contributes 2^{i+j} to $\sum A_{uv}^2$

$\times 2$ if $i \neq j$ and we only consider $i \leq j$

$\boxed{A}$

Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\quad n \leq 10^5$

$\underline{\sum A_{uv}^2}$

If i-th and j-th bit are set in A_{uv},

it contributes 2^{i+j} to $\sum A_{uv}^2$

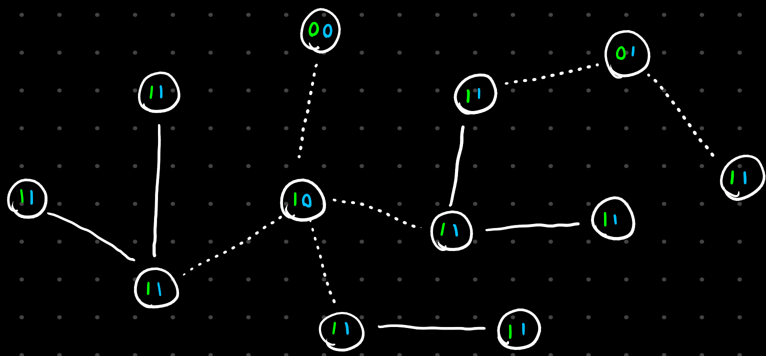
Count the # of paths that have the i-th and j-th set

A

Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\qquad n \leq 10^5$

$\underline{\sum A_{uv}^2}$

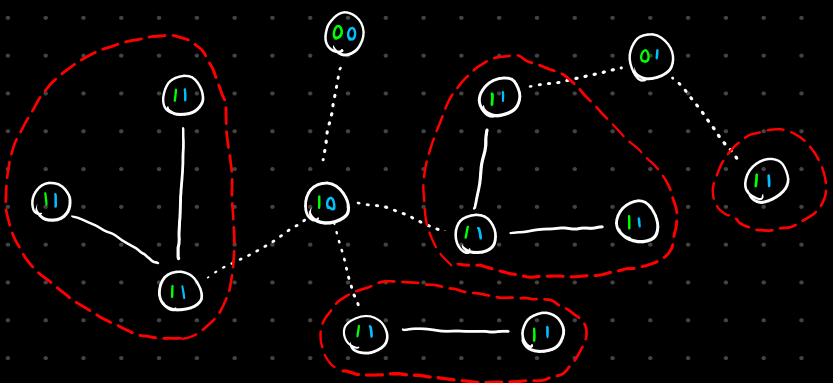


$\boxed{A}$

Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\quad n \leq 10^5$

$\sum A_{uv}^2$



Count # of paths in each red component

$\boxed{A}$

Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\qquad n \leq 10^5$

$\underline{\sum O_{uv}^2}$ Assume every bit is set initially.

Subtract the contribution of paths where neither i nor j is ever set

$\rightarrow$ symmetric to AND

$\boxed{A}$

Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\qquad n \leq 10^5$

$\sum X_{uv}^2$

For each i, j do a DP.

$dp[u][a][b]$ – # of paths from subtree of u to u s.t.

a – is the i-th bit set in the XOR?
b – is the j-th bit set in the XOR?

$\boxed{A}$ Given a tree, every vertex has a value $< 2^{25}$
Let A_{uv}, O_{uv}, X_{uv} be the AND/OR/XOR of the path $u \sim v$.

Find $\sum A_{uv}^2$, $\sum O_{uv}^2$, $\sum X_{uv}^2$ $\quad n \leqslant 10^5$

$\underline{\sum X_{uv}^2}$

For each i, j do a DP.

dp[u][a][b] – # of paths from subtree of u to u s.t.

a – is the i-th bit set in the XOR?
b – is the j-th bit set in the XOR?

At every u, update the answer for paths with LCA at u.

$\boxed{D}$ Given a binary string, count the # of distinct subsequences of length k.

$n \leq 2 \cdot 10^5$

$\boxed{D}$

Given a binary string, count the # of distinct subsequences of length k.

$n \leq 2 \cdot 10^5$

Always consider the "leftmost" occurrence of a subsequence.

0 0 1 0 1 1 0 1 0 0

"canonical occurrence"

$\boxed{D}$ Given a binary string, count the # of distinct subsequences of length k.

$n \leq 2 \cdot 10^5$

Divide & conquer

Split the array in 2 roughly equal pieces

Calculate # of subsequences in each, for each subseq length

Merge the results back

How to ensure only "canonical" occurrences are counted?

$\boxed{D}$ Given a binary string, count the # of distinct subsequences of length k.

$n \leq 2 \cdot 10^5$

How to ensure only "canonical" occurrences are counted?

Maintain for each counted subsequence

- its first character
- whether it can be expanded inside the current subarray by adding a '0'
- whether it can be expanded inside the current subarray by adding a '1'

0 1 1 0 1 1 0 0 1 0	0

D

Given a binary string, count the # of distinct subsequences of length k.

$n \leq 2 \cdot 10^5$

$dp[a][b][k]$ – # of subsequences

$a \in \{0, 1\}$ starting with a

$b \in \{0, 1, 2\}$ that can't be extended by adding b

$0 \leq k \leq$ length with length k

0 1 1 0 1 1 0 0 1 0	0

Counts for $dp[0][1][7]$, $dp[0][2][7]$

but not $dp[0][0][7]$

D

Given a binary string, count the # of distinct subsequences of length k.

$n \leq 2 \cdot 10^5$

If we have dp values for two subarrays, how to merge?

$$dp[0][2][k] = \sum_{i+j=k} dp_L[0][1][i] \cdot dp_R[1][2][j]$$

$$+ \sum_{i+j=k} dp_L[0][0][i] \cdot dp_R[0][2][j] \quad \text{etc}$$

Use FFT. May need some special care for $k=0$

D

Given a binary string, count the # of distinct subsequences of length k.

$n \leq 2 \cdot 10^5$

If we have dp values for two subarrays, how to merge?

$$dp[0][2][k] = \sum_{i+j=k} dp_L[0][1][i] \cdot dp_R[1][2][j]$$

$$+ \sum_{i+j=k} dp_L[0][0][i] \cdot dp_R[0][2][j] \quad \text{etc}$$

Use FFT. May need some special care for $k = 0$

$$T(n) = 2T(n/2) + O(n \log n) \rightarrow O(n \log^2 n)$$

$\boxed{C}$

Given a permutation, you may use this operation any # of times:

- Pick $k > 0$ indices $x_1 < x_2 < \cdots < x_k$ and cyclically shift them by 1. Cost $1/k$.

Sort the array with cost ≤ 2.

$\boxed{C}$

Given a permutation, you may use this operation any # of times:

- Pick $k > 0$ indices $x_1 < x_2 < \cdots < x_k$ and cyclically shift them by 1. Cost $1/k$.

Sort the array with cost ≤ 2.

Small k bad, big k good.

$k = n$ is not very useful.

What about $k = n - 1$?

a b c d e f g h

$\downarrow$

(h) a b c e d f g h

Swaps the unpicked element with previous and rotates.

$\boxed{C}$

Given a permutation, you may use this operation any # of times:

- Pick $k > 0$ indices $x_1 < x_2 < \cdots < x_k$ and cyclically shift them by 1. Cost $1/k$.

Sort the array with cost ≤ 2.

Similar thing happens whenever no unpicked elements are adjacent

$\underline{a}$ b $\underline{c}$ $\underline{d}$ e $\underline{f}$ g $\underline{h}$

(h) b a c e d g f h

$\boxed{C}$

Given a permutation, you may use this operation any # of times:

- Pick $k > 0$ indices $x_1 < x_2 < \cdots < x_k$ and cyclically shift them by 1. Cost $1/k$.

Sort the array with cost ≤ 2.

Similar thing happens whenever no unpicked elements are adjacent

a b c d e f g h

(h) b a c e d g f h

$\rightarrow$ Can swap any set of non-intersecting pairs of elements.

$$\text{Cost} \leq \frac{1}{n/2} = \frac{2}{n}$$

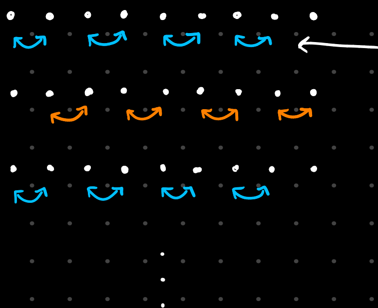
C

Given a permutation, you may use this operation any # of times:

- Pick $k > 0$ indices $x_1 < x_2 < \cdots < x_k$ and cyclically shift them by 1. Cost $1/k$.

Sort the array with cost ≤ 2.

Odd-even sort



make the swaps in the pairs where order is wrong

Can show: needs n passes.

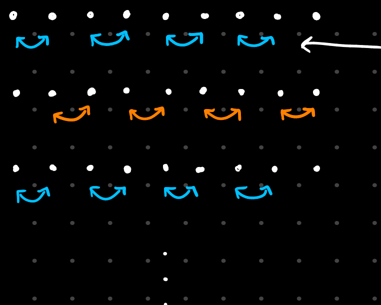
$\boxed{C}$

Given a permutation, you may use this operation any # of times:

- Pick $k > 0$ indices $x_1 < x_2 < \cdots < x_k$ and cyclically shift them by 1. Cost $1/k$.

Sort the array with cost ≤ 2.

Odd - even sort



make the swaps in the pairs where order is wrong

Can show: needs n passes.

$\rightarrow$ cost 2
rotation cancels out with n passes

$\boxed{H}$

Given a DAG. Each vertex has a capacity for gifts. When a vertex reaches this capacity it gives 1 gift to each outgoing edge and throws the rest away, potentially triggering a chain reaction.

Each day a new gift is given to vertex 1. What is the first day no one has any gifts?

$n \leqslant 10^4 \quad m \leqslant 3 \cdot 10^4$

capacities $\leqslant 10^5$

H

Given a DAG. Each vertex has a capacity for gifts. When a vertex reaches this capacity it gives 1 gift to each outgoing edge and throws the rest away, potentially triggering a chain reaction.

Each day a new gift is given to vertex 1. What is the first day no one has any gifts?

$n \leq 10^4 \quad m \leq 3 \cdot 10^4$
capacities $\leq 10^5$

The order of regifting doesn't actually matter.

H

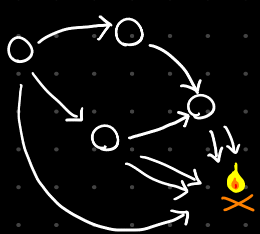
Given a DAG. Each vertex has a capacity for gifts. When a vertex reaches this capacity it gives 1 gift to each outgoing edge and throws the rest away, potentially triggering a chain reaction.

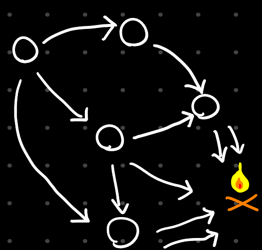
Each day a new gift is given to vertex 1. What is the first day no one has any gifts?

$n \leq 10^4$ $m \leq 3 \cdot 10^4$
capacities $\leq 10^5$

Try to build the answer by adding vertices in topo order

(i.e. pretend that vertices $i+1 \dots n$ are incinerators)



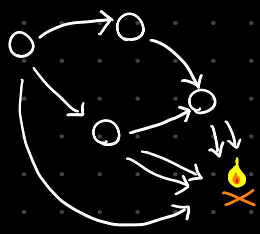


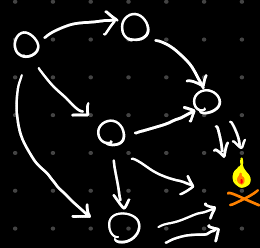
H

$n \leq 10^4$ $m \leq 3 \cdot 10^4$
capacities $\leq 10^5$

Try to build the answer by adding vertices in topo order

(i.e. pretend that vertices $i+1 \ldots n$ are incinerators)





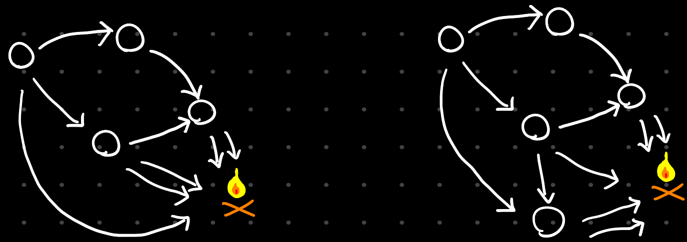
Induction hypothesis – state of first i vertices is periodic
dp[i] – period

H

$n \leq 10^4 \quad m \leq 3 \cdot 10^4$
capacities $\leq 10^5$

Try to build the answer by adding vertices in topo order

(i.e. pretend that vertices $i+1 \dots n$ are incinerators)



Induction hypothesis – state of first i vertices is periodic
$dp[i]$ – period

Try to add vertex $i+1$. Gets S gifts every period.

$$\rightarrow \text{ new period is } \quad dp[i] \cdot \frac{capacity[i+1]}{\gcd(capacity[i+1],\ S)}$$

$\boxed{H}$

$n \leq 10^4$, $m \leq 3 \cdot 10^4$
capacities $\leq 10^5$

Try to add vertex $i+1$. Gets S gifts every period.

$\rightarrow$ new period is $dp[i] \cdot \dfrac{capacity[i+1]}{\gcd(capacity[i+1],\ S)}$

How to calculate S?

$\boxed{H}$

$n \le 10^4 \quad m \le 3 \cdot 10^4$
capacities $\le 10^5$

Try to add vertex $i+1$. Gets S gifts every period.

$\rightarrow$ new period is $dp[i] \cdot \dfrac{capacity[i+1]}{\gcd(capacity[i+1],\ S)}$

How to calculate S?

Maintain $gifts[j]$ – # of gifts j receives during its period

Contributes $\dfrac{dp[i] \cdot gifts[j]}{dp[j] \cdot capacity[j]}$ to S if $j \to i+1$ exists

$\boxed{E}$

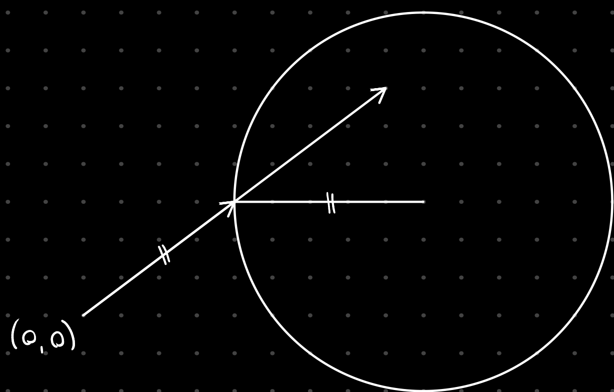
Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form:

- Given a ray from the origin and $k \leq 5$. You move at speed 1 along that ray. What is the first time we are attacked by $\geq k$ points?

$n \leq 10^5$, $q \leq 10^5$

$\boxed{E}$ Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form above.

When are we first attacked by a given point?

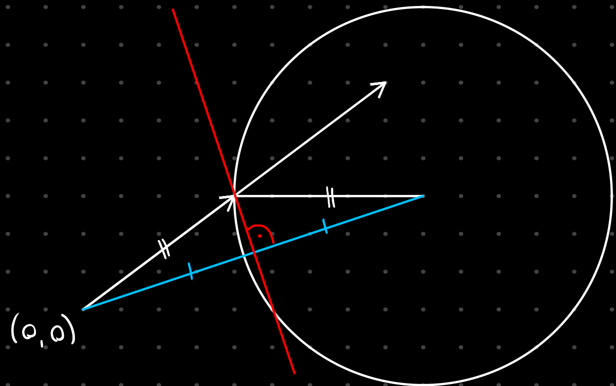


Once the distance between $(0,0)$ and the infection point is equal.

E

Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form above.

When are we first attached by a given point?



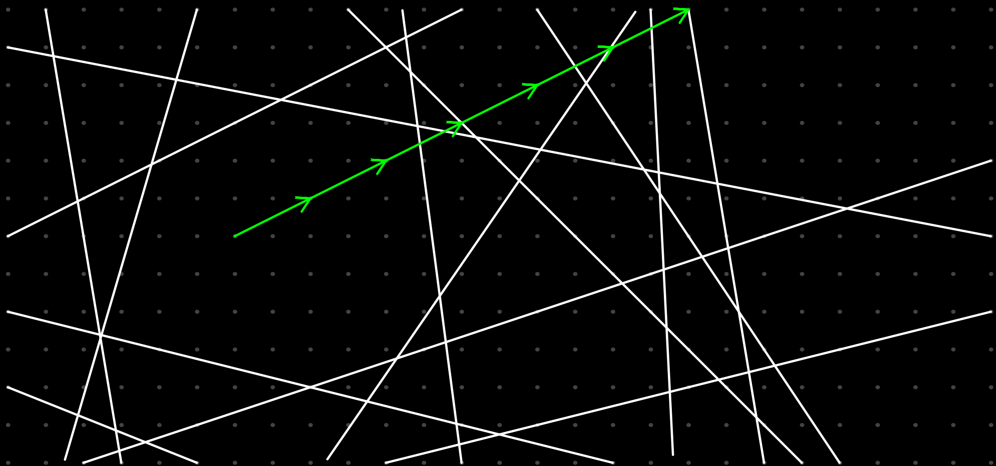
Location of red line doesn't depend on direction.

Once we are attached, we never stop being attached.

Once the distance between (0,0) and the infection point is equal.
i.e. once we cross the red line.

E

Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form above.

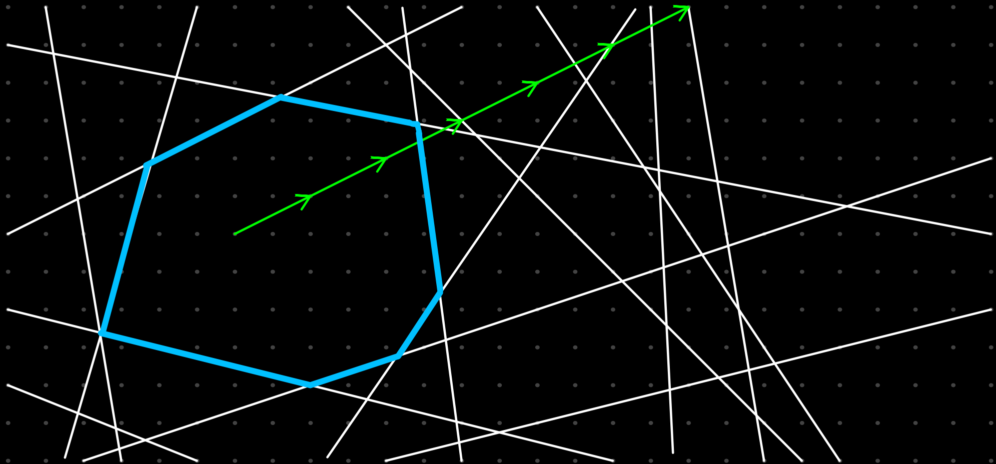


Calculate all those lines.

Each query is "when is the k-th line we cross"?

$\boxed{E}$

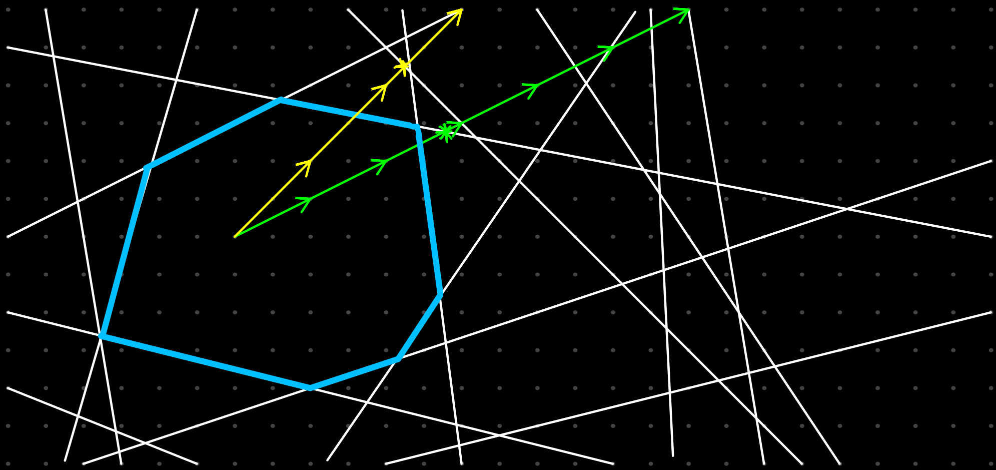
Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form above.



For $k=1$, answer is the time of crossing the blue polygon.

$\boxed{E}$

Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form above.

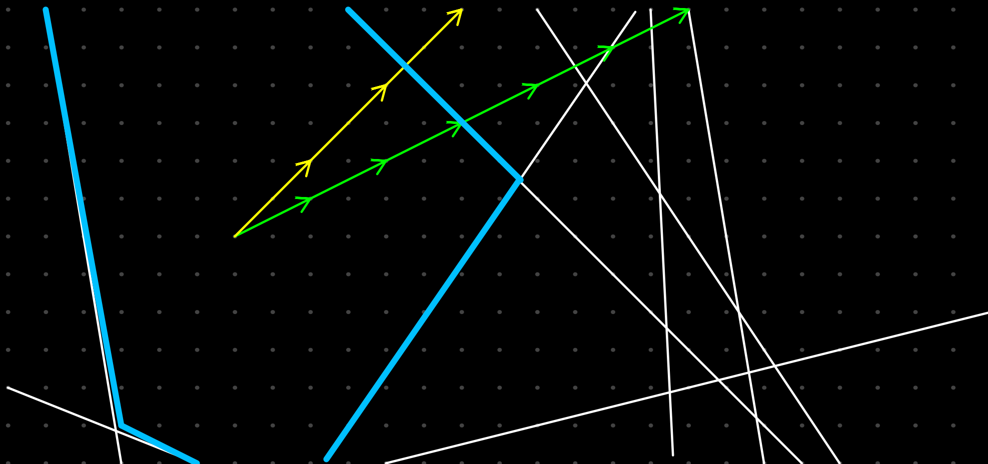


For $k=2$, the answer is either a line neighboring the line crossed in the polygon

Or a line that is not in the polygon at all.

$\boxed{E}$

Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form above.



Delete all lines in the blue polygon and find the new one.

Repeat.

$\boxed{E}$ Given n points (x_i, y_i). At time t, you are under attack by the i-th point iff the distance to it is $\leq t$. Solve q queries of the form above.

To answer a query:

- calculate the intersection of the ray with each polygon

- take that line and 4 neighboring lines on each side

- calculate the intersection of the ray with all picked lines

- choose the k-th minimum

$\boxed{B}$

A subarray of a permutation is <u>unstable</u> if its maximum is its first or last element.

A permutation is called <u>balanced</u> if it has the minimum # of balanced subarrays.

Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \le 10^5$, $k, l \le 10^{18}$

$\boxed{B}$

A subarray of a permutation is <u>unstable</u> if its maximum is its first or last element.

A permutation is called <u>balanced</u> if it has the minimum # of balanced subarrays.

Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \leq 10^5, \; k, l \leq 10^{18}$

What do balanced permutations look like?

$\boxed{B}$

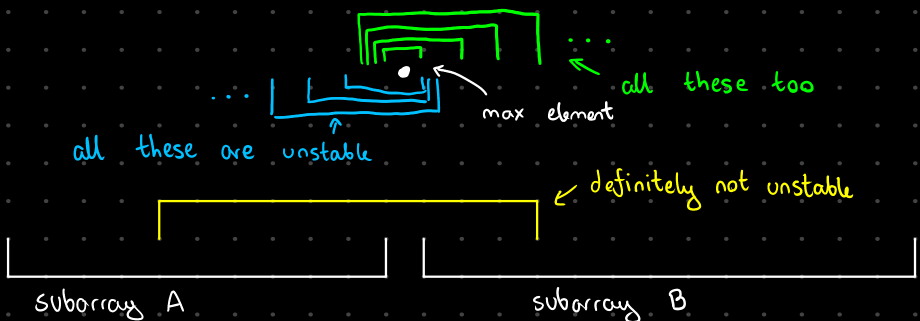
A subarray of a permutation is <u>unstable</u> if its maximum is its first or last element.

A permutation is called <u>balanced</u> if it has the minimum # of balanced subarrays.

Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \le 10^5, \ k, l \le 10^{18}$

What do balanced permutations look like?

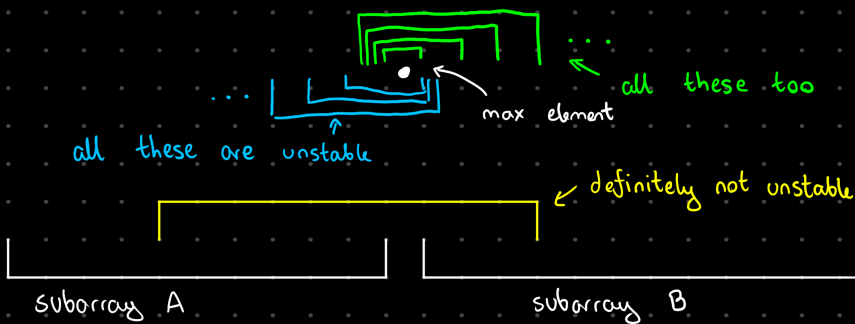


B

Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \le 10^5$, $k, l \le 10^{18}$

What do balanced permutations look like?

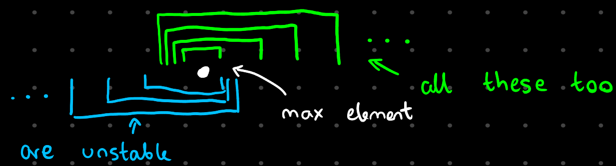


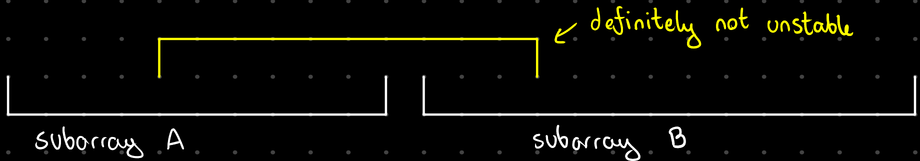
unstables(p) = unstables(A) + unstables(B) + n

$\boxed{B}$ Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \leq 10^5$, $k, l \leq 10^{18}$

What do balanced permutations look like?



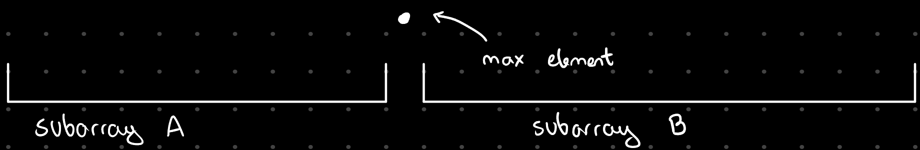


$$\text{unstables}(p) = \text{unstables}(A) + \text{unstables}(B) + n$$

$\boxed{B}$ Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \le 10^5$, $k, l \le 10^{18}$

What do balanced permutations look like?



$\text{unstables}(p) = \text{unstables}(A) + \text{unstables}(B) + n$

To build a balanced perm:

use brute force to find the pattern for allowed range (it is not just 1 or 2)

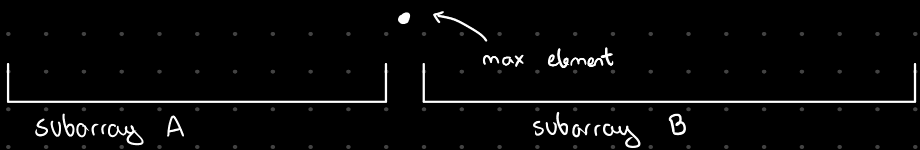
- put the max element "somewhere in the middle"
- choose a balanced perm for either side, uncompress somehow

$\binom{n-1}{\ell}$ ways

$\boxed{B}$ Find the k-th lex-min and ℓ-th lex-max balanced permutation of length n.

$n \le 10^5$, $k, \ell \le 10^{18}$

What do balanced permutations look like?



unstables(p) = unstables(A) + unstables(B) + n

To build a balanced perm:

use brute force to find the pattern for allowed range (it is not just 1 or 2)

- put the max element "somewhere in the middle"
- choose a balanced perm for either side, uncompress somehow

$\binom{n-1}{\ell}$ ways

B

Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \le 10^5, \; k, l \le 10^{18}$

<u>Lex-min</u> For large n, the left half must definitely be lex-min, so we can shorten the array quite a bit.

B

Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \le 10^5$, $k, l \le 10^{18}$

Lex-min For large n, the left half must definitely be lex-min, so we can shorten the array quite a bit.

Assume $n \le 40$. Standard lex-min strategy:

- Learn to answer queries like "count # of balanced permutations with given prefix"

- Construct the answer greedily

$\boxed{B}$ Find the k-th lex-min and l-th lex-max balanced permutation of length n.

$n \le 10^5$, $k, l \le 10^{18}$

$n \le 40$. How many balanced permutations have given prefix?

Some $O(n^4)$ or $O(n^5)$ DP that considers every subarray and every maximum value for every subarray.