

ECUST Programming Contest 2026 Fall

Official Contest – Problem Statements

September 29, 2026, 13:00–18:00

Problem List

Problem	Title	Limits	Note
A	Synecdoche, New York	2s/1GB	
B	AIMC Lab	2s/1GB	
C	Split in Two 2	2s/1GB	
D	Unit Distance!?	2s/1GB	
E	Eradication	4s/1GB	
F	Convex Hull	2s/1GB	
G	Beyond the Stars	8s/1GB	Communication
H	Ordinary Offensive Magic	2s/1GB	
I	Shooting Stars	1s/1GB	
J	Time Trouble 2	1s/1GB	
K	Aeonglass	2s/1GB	
L	In Filtration 3	2s/1GB	
M	Witch Trial	6s/1GB	

This booklet contains 13 problems. If pages are missing, contact a volunteer immediately.

Please read the contest rules and technical rules included with the contest files.

Do not open the booklet or touch the mouse or keyboard before the contest begins.

Black-and-white printing may obscure information conveyed by color.

The italic text before each problem statement is flavor text and is unrelated to the problem. It is intentionally left untranslated.

The English version is translated by GPT 6.1 Sol, In case of any discrepancies, the Chinese problem statement shall prevail.

Problem **A** Synecdoche, New York

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

And they say there is no fate, but there is, it's what you create.

And even though the world goes on for eons and eons...

...you are only here for a fraction of a fraction of a second.

Problem Statement

Caden has come to understand “fate” in his own way. Leaving theater behind, he wants to use a string to represent his life through synecdoche. But in seeking a pure, original expression true to himself, he again falls into an infinite loop of self-reference. This time, Caden leaves the choice to you: portray his life for him.

Caden’s synecdoche is a string T of length L . We call T **valid** if and only if all three conditions below hold:

1. T consists only of lowercase English letters.
2. For every $i = 1, 2, \dots, K$, the character of T at position p_i must equal the given character c_i , where $1 \leq p_i \leq L$ and c_i is a lowercase letter.
3. For every $i = 1, 2, \dots, N$, the given string s_i must **not** occur as a contiguous substring of T .

Given L , K position constraints (p_i, c_i) , and N forbidden substrings s_i , count all valid strings T . Output the answer modulo 998 244 353.

Input

The first line contains three integers L , K , and N ($1 \leq L \leq 10^{18}$, $0 \leq K \leq 500$, $0 \leq N \leq 50$): the string length, the number of position constraints, and the number of forbidden substrings.

Each of the next K lines contains an integer p_i and a lowercase letter c_i ($1 \leq p_i \leq L$), describing a position constraint. All p_i are distinct.

Each of the next N lines contains a nonempty string s_i , describing a forbidden substring.

The sum of the lengths of all s_i does not exceed 100; that is, $\sum_{i=1}^N |s_i| \leq 100$.

Output

Output one integer for each test case: the number of valid strings modulo 998 244 353.

Sample 1

standard input	standard output
<pre>3 1 1 2 b ab</pre>	<pre>650</pre>

Sample 2

standard input	standard output
<pre>10 2 2 3 a 5 c abc dcabe</pre>	<pre>113392371</pre>

Sample 1 Explanation

The string has length 3, its second character must be **b**, and it must not contain **ab**. Valid strings have the form **?b?**; the first character can be any lowercase letter except **a**. There are $25 \times 26 = 650$ such strings.

Problem B AIMC Lab

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

街灯璀璨，可我日渐透明。

Problem Statement

Your AIMC Lab has developed an entirely new three-dimensional quantum matrix!

Consider a three-dimensional grid of size $N \times M \times H$, with coordinates $1 \leq x \leq N$, $1 \leq y \leq M$, and $1 \leq z \leq H$. A qubit occupies every integer grid point. Let (x_u, y_u, z_u) be the coordinates of qubit u .

For any two qubits u and v , define their **squared spatial distance** as

$$d^2(u, v) = (x_u - x_v)^2 + (y_u - y_v)^2 + (z_u - z_v)^2.$$

The system has eight energy matrices $A_{D,c}$, each of size $N \times M \times H$, indexed by integers $0 \leq D \leq 3$ and $0 \leq c \leq 1$.

You may initialize each qubit to either basis state $|0\rangle$ or $|1\rangle$. Its corresponding basis-state value is $c_P = 0$ or $c_P = 1$.

After initialization, the total energy is calculated as follows:

- For each integer pair $0 \leq D \leq 3$, $0 \leq c \leq 1$, qubit P gains $A_{D,c}(x_P, y_P, z_P)$ energy if and only if:
 - The basis-state value of P satisfies $c_P = c$.
 - For every qubit Q with $d^2(P, Q) = D$, the value $c_P + c_Q + D$ is even.

As the chief scientist of AIMC Lab, initialize every qubit to a basis state so that the system's total energy is maximized. Output this maximum energy.

Input

The first line contains three integers N , M , and H ($1 \leq N, M, H \leq 100$ and $N \times M \times H \leq 100$), the grid dimensions.

The data for the eight energy matrices $A_{D,c}$ follow in lexicographic order of (D, c) .

Each matrix consists of N blocks. Each block has M lines, each containing H integers. Blocks are separated by blank lines.

The z -th number in the y -th line of the x -th block is the energy at (x, y, z) . All energy values satisfy $A_{D,c}(x_P, y_P, z_P) \in [0, 10^9]$.

Output

Output one integer: the maximum total energy the system can achieve.

Sample 1

standard input	standard output
1 1 1	2265
3	
5	
100	
200	
1000	
2000	
50	
60	

Sample 1 Explanation

The grid contains only one qubit, at $(1, 1, 1)$.

Initializing it to $|1\rangle$ gives $A_{0,1}(1, 1, 1) + A_{1,1}(1, 1, 1) + A_{2,1}(1, 1, 1) + A_{3,1}(1, 1, 1) = 5 + 200 + 2000 + 60 = 2265$, which is optimal.

Sample 2

standard input	standard output
2 2 2	88
1 1	
1 1	
1 1	
1 1	
1 1	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
10 10	
10 10	
10 10	
10 10	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	

Sample 2 Explanation

Initialize all 8 qubits of the $2 \times 2 \times 2$ grid to $|0\rangle$.

- Each qubit gains 1 energy from $A_{0,0}$, for a total of 8.
- All qubits at squared distance 2 are also in state $|0\rangle$, so each qubit gains 10 from $A_{2,0}$, for a total of 80.

The total energy is $8 + 80 = 88$. This can be shown to be the maximum.

Problem C Split in Two 2

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

别在昨日里频频回首，说什么将寂寞对半分担。

Problem Statement

You are given two arrays a and b , each of length n .

Choose any $k \in [0, n]$ and concatenate the first k elements of a with the last $n - k$ elements of b to form a new array of length n .

Select exactly m pairwise nonadjacent elements from the new array. Let p be the smallest selected element. Maximize p .

Input

The first line contains an integer T ($1 \leq T \leq 10^4$), the number of test cases.

The first line of each test case contains two integers n and m ($1 \leq m \leq \lceil \frac{n}{2} \rceil \leq n \leq 2 \times 10^5$): the array length and the number of elements to select.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$).

The third line contains n integers $b_1, b_2, \dots, b_n$ ($-10^9 \leq b_i \leq 10^9$).

The sum of n over all test cases does not exceed 2×10^5 .

Output

For each test case, output one integer: the maximum possible value of p .

Sample 1

standard input	standard output
2	5
5 2	-4
5 4 3 2 1	
1 2 3 4 5	
9 5	
-1 1 -4 -5 -1 4 -1 -9 1	
-9 -9 -8 2 4 -4 -3 -5 3	

Sample 1 Explanation

In the first test case, choosing $k = 4$ gives $[5, 4, 3, 2, 5]$. Select the first and fifth elements; their minimum is 5, which is optimal.

Problem **D** Unit Distance!?

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

最后跳支舞吧。
在魔都的远郊，
在化工强校的信息楼，
在这个光怪陆离的
革命前夜。

Problem Statement

Construct a set of n points in the Euclidean plane such that:

- The absolute values of both coordinates of every point are at most 10^5 ;
- Each coordinate of every point becomes an integer when multiplied by 10^{13} ;
- The distance between any two points is at least 10^{-4} ;
- At least $\lfloor \frac{1}{2}n \log_2 n - n \rfloor$ distinct unordered pairs of points have distance in $[1 - 10^{-9}, 1 + 10^{-9}]$.

Input

The first line contains an integer T ($1 \leq T \leq 100$), the number of test cases.

Each test case contains one integer n ($2 \leq n \leq 5000$), the number of points to construct.

The sum of n over all test cases does not exceed 5000.

Output

For each test case, output n lines. The i -th line contains two integers X_i, Y_i ($-10^{18} \leq X_i, Y_i \leq 10^{18}$), representing the actual point $(\frac{X_i}{10^{13}}, \frac{Y_i}{10^{13}})$.

Your output is accepted if and only if it satisfies this format and all four conditions above.

If several valid constructions exist, output any one of them.

Sample 1

standard input	standard output
<pre>1 3</pre>	<pre>0 0 10000000000000 0 5000000000000 8660254038000</pre>

Sample 1 Explanation

In the first test case, the actual coordinates are:

$$(x_1, y_1) = (0, 0), (x_2, y_2) = (1, 0), (x_3, y_3) = (0.5, 0.8660254038)$$

The first three conditions are easy to verify. The pairwise distances are:

$$d_{12} = 1, d_{13} = d_{23} = 1.000000000013476527219909191607 \dots \approx 1 + 1.3476 \times 10^{-11}$$

Thus every pairwise distance is within the allowed tolerance of 1, satisfying the fourth condition.

Problem E Eradication

Time limit 4 seconds
Memory limit 1 Gigabyte
Input standard input
Output standard output

我们终于等来了黎明，可这座城的根除了记得黑夜，似乎什么也没能留下。

Problem Statement

We often simplify the square root of a large integer into radical form; for example, $\sqrt{432} = 12\sqrt{3}$. Define the number-theoretic function $f(n) = \min\{a \mid n = ab^2, a, b \in \mathbb{N}\}$ on $\mathbb{N}$. In other words, $f(n)$ is the smallest natural number a satisfying the equation. Such an a always exists. You are given q queries, each specifying l and r . For each query, find $\sum_{i=l}^r f(i)$ modulo 998 244 353.

Input

The first line contains an integer q ($1 \leq q \leq 500$), the number of queries. Each of the next q lines contains two integers l and r ($1 \leq l \leq r \leq 10^{12}$), specifying a query interval.

Output

For each query, output one integer: $\sum_{i=l}^r f(i)$ modulo 998 244 353.

Sample 1

standard input	standard output
4	3
432 432	38
1 10	66
10 15	651260790
1 1000000000	

Sample 1 Explanation

For the second and third queries, the corresponding values of f are:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(n)$	1	2	3	1	5	6	7	2	1	10	11	3	13	14	15

In particular, the third query gives

$$\sum_{i=10}^{15} f(i) = f(10) + f(11) + f(12) + f(13) + f(14) + f(15) = 10 + 11 + 3 + 13 + 14 + 15 = 66.$$

Problem **F** Convex Hull

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

我知道既不是凸包也不是凹包的我为什么要哭的。因为我其实是一只豆包。

Problem Statement

You are given n quadratic inequalities, each of the form

$$a_i x^2 + b_i x + c_i > 0$$

It is guaranteed that $a_i < 0$ for every $1 \leq i \leq n$.

Determine whether there is an **integer** x satisfying all n inequalities simultaneously.

Input

The first line contains an integer T ($1 \leq T \leq 10^4$), the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 2 \times 10^5$), the number of inequalities.

Each of the next n lines contains three integers a_i, b_i, c_i ($-10^5 \leq a_i \leq -1$, $-10^5 \leq b_i, c_i \leq 10^5$), the coefficients of the i -th inequality.

The sum of n over all test cases does not exceed 2×10^5 .

Output

For each test case, output **YES** if such an integer x exists, or **NO** otherwise.

The answer is case-insensitive; for example, **yes**, **YES**, **Yes**, and **YeS** are all accepted as affirmative answers.

Sample 1

standard input	standard output
5	NO
2	YES
-1 0 4	NO
-1 4 -3	YES
3	NO
-1 100000 0	
-1 202 10000	
-2 400 19998	
2	
-1 2 0	
-1 0 -1	
2	
-100000 0 100000	
-1 0 4	
2	
-1 2 0	
-1 4 -3	

Sample 1 Explanation

In the fourth test case, $x = 0$ satisfies both inequalities, so an integer solution exists.

Problem **G** Beyond the Stars

Time limit 8 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

今夜天空深远宁静，我们望的是同一颗星星吗？

This is a communication problem. Pay attention to its unusual constraints.

Problem Statement

That was long ago, in a past forgotten by all of ECUST...

In that forgotten age, Zhulin and Qianwai fought side by side, lived together, and gazed up at the same stars. Later, sand buried their language, oceans severed their roads, and even their names faded from history. Only the vast, silent star chart above them remained: public, radiant, and unchanging. Yet nobody knew which stars they had once quietly joined into a message.

This problem has 20 test cases.

You are given an undirected graph $G = (V, E)$ with $|V| = 2000$ and $|E| = 43\,600$. It has no multiple edges or self-loops and is generated as follows. Initially, G has $|V|$ vertices and no edges. Then, $|E|$ times, the generator uniformly chooses one of the vertex pairs not yet joined by an edge and adds an undirected edge between them.

Your program will be run twice:

First run

In the first run, you play Zhulin. Choose and output n induced four-cycles in G . They may share vertices.

- **Definition of an induced four-cycle:** Choose 4 vertices of G that have **exactly** 4 edges among them, forming a simple cycle. In particular, there are no diagonals.

Second run

In the second run, you play Qianwai. The judge has your complete output from the first run. For each of q queries, it selects one of your four-cycles, deletes one of its vertices, and gives you the other three vertex numbers. Report the index of the four-cycle in your first-run output from which they came.

These three vertices may belong to several of the four-cycles you chose, but you must report the one selected by the judge.

Input

First run

The first line contains the string **send**.

The second line contains an integer n ($n = 200\,000$), the number of induced four-cycles you must output.

The third line contains two integers $|V|$ and $|E|$.

Each of the next $|E|$ lines contains two integers u, v , indicating an undirected edge between vertices u and v ($1 \leq u, v \leq |V|$).

Second run

The first line contains the string **receive**.

The second line contains two integers $|V|$ and $|E|$.

Each of the next $|E|$ lines contains two integers u, v , indicating an undirected edge between vertices u and v ($1 \leq u, v \leq |V|$).

Line $|E| + 3$ contains an integer q ($q = 4n$), the number of received vertex sets.

Each of the next q lines contains 3 integers: the vertices remaining after the judge removes one vertex from a four-cycle in your first-run output.

Output

First run

Output n lines. Line i contains four integers v_1, v_2, v_3, v_4 , the vertices of the i -th induced four-cycle.

Second run

Output q lines. For each received vertex set, output an integer i ($1 \leq i \leq n$), the index of its four-cycle.

Sample 1

First run

standard input	standard output
send	1 174 613 1323
200000	1 955 1683 1755
2000 43600	...
29 1973	1390 1615 1683 1777
228 1127	
...	
92 573	

Second run

standard input	standard output
receive	17713
2000 43600	191947
29 1973	...
228 1127	118231
...	
92 573	
800000	
46 1297 1949	
1056 1275 1454	
...	
387 1289 1533	

Sample 1 Explanation

Parts of the sample are omitted because the complete data is too large. The complete input for the first run is available in the attachments.

Problem H Ordinary Offensive Magic

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

“Zoltraak.”

Problem Statement

Consider an $n \times m$ grid. Each cell has one of three states:

1. $/$ (forward-slash mirror): reflects on both sides. A ray turns 90° , connecting the top with the left and the bottom with the right.
2. $\backslash$ (backslash mirror): reflects on both sides. A ray turns 90° , connecting the top with the right and the bottom with the left.
3. $.$ (empty cell): contains no obstacle. A ray passes straight through, connecting the top with the bottom and the left with the right.

Initially, some cells contain fixed $/$ or $\backslash$ mirrors; the rest are empty. You may **leave each initially empty cell empty, replace it with a $/$ mirror, or replace it with a $\backslash$ mirror.**

The outside boundary of the grid consists of $2(n + m)$ unit-length segments. A ray may enter or leave only perpendicularly through the midpoint of one of these segments.

Given an entry point and a target exit point, determine whether the grid can be filled legally so that a ray entering at the specified point, after straight-line travel and reflections, exits at exactly the target point.

If possible, output a valid completed grid that **minimizes the number of cells traversed** by the ray from the entry point to the target exit point. Count one traversal each time the ray enters a cell; repeated visits to the same cell are counted separately. If there are multiple optimal completions, output any of them. Otherwise, report that it is impossible.

Input

The first line contains an integer T ($1 \leq T \leq 100$), the number of test cases.

The first line of each test case contains two integers n, m ($1 \leq n \times m \leq 2 \times 10^5$), the numbers of rows and columns.

Each of the next n lines contains a string of length m describing the initial grid. Here $/$ is a forward-slash mirror, $\backslash$ is a backslash mirror, and $.$ is an empty cell.

The next line contains four space-separated values d_1, p_1, d_2, p_2 . The first pair specifies the entry point, and the second specifies the target exit point:

- Each of d_1, d_2 is one of U, D, L, R, denoting the top, bottom, left, or right boundary, respectively.
- Each of p_1, p_2 is a positive integer specifying the midpoint of the outer edge of the p -th boundary cell. On U or D, p is a column number ($1 \leq p \leq m$); on L or R, it is a row number ($1 \leq p \leq n$).

It is guaranteed that $(d_1, p_1) \neq (d_2, p_2)$.

The sum of $n \times m$ over all test cases does not exceed 2×10^5 .

Output

For each test case, if a valid completion exists, output n lines of length m describing a completion that minimizes the number of cells traversed. If there are multiple optimal completions, output any of them.

Do not output the minimum number of traversals. Otherwise, output **IMPOSSIBLE**.

A backslash must usually be escaped: `\\` outputs $\backslash$.

Sample 1

standard input

```
3
2 2
..
..
L 1 U 2
4 4
^ ^
^ /.
....
./..
L 2 L 4
4 4
^ ^
^ /.
.. \ /
./..
L 2 L 4
```

standard output

```
./
..
^ ^
^ //
.//.
./..
IMPOSSIBLE
```

Sample 1 Explanation

In the first test case, the ray enters from the left edge of row 1 and travels right. It passes through the empty cell (1, 1), turns upward at the / mirror in (1, 2), and exits through the top edge of column 2. This completion traverses 2 cells, which is the minimum possible.

Problem **I** Shooting Stars

Time limit 1 second

Memory limit 1 Gigabyte

Input standard input

Output standard output

在燥热的季节，一颗流星划过天际，留下光亮的拖尾。

我循着光奔跑的那个夜晚，便是故事的开端。

Problem Statement

There are n crystals numbered 1 through n from left to right. Crystal i initially contains a_i units of energy. You may collect all n crystals in any order.

When you collect crystal i :

- You add its current energy to your total score.
- If $i > 1$ and crystal $i - 1$ has not yet been collected, its current energy changes sign.
- If $i < n$ and crystal $i + 1$ has not yet been collected, its current energy changes sign.

The crystals' numbers do not change as you collect them.

What is the maximum total score after collecting all n crystals?

Input

The first line contains an integer T ($1 \leq T \leq 10^4$), the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 2 \times 10^5$), the number of crystals.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$), their initial energies.

The sum of n over all test cases does not exceed 2×10^5 .

Output

For each test case, output one integer: the maximum possible total score.

Sample 1

standard input	standard output
2	15
5	47
1 2 3 4 5	
9	
9 -9 8 -2 4 -4 3 -5 3	

Sample 1 Explanation

In the first test case, collecting crystals in the order $3 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 4$ gives a score of 15. This is optimal.

Problem **J** Time Trouble 2

Time limit 1 second

Memory limit 1 Gigabyte

Input standard input

Output standard output

“对无数时间线建立可持久化字典树即可实现对历史的版本控制”，一位青铜龙朋友总是这么告诉我。我对此深信不疑，以至于忘记了自己倒数的心跳。

Problem Statement

To prove that he is never late for class, Xingyu insists that he entered the classroom at 7:110 for a class beginning at 8:00. This inspires a “generalized time notation.”

In generalized time notation, a time is written $A:B$, where A, B are nonnegative integers.

We define two **equivalence rules** for generalized times:

1. For any $B \geq 60$, $A:B$ and $(A+1):(B-60)$ represent the same instant.
2. For any $A \geq 24$, $A:B$ and $(A-24):B$ represent the same instant.

A time $h:m$ is a “standard time” if and only if $0 \leq h < 24$ and $0 \leq m < 60$.

Given a generalized time, find the unique equivalent standard time.

Input

The first line contains an integer T ($1 \leq T \leq 10^5$), the number of test cases.

Each test case contains two integers A, B ($0 \leq A, B \leq 10^9$), representing the generalized time $A:B$.

Output

For each test case, output two integers h, m representing the equivalent standard time $h:m$.

Sample 1

standard input	standard output
5	8 50
7 110	1 30
25 30	0 0
0 0	23 59
23 59	7 20
100 200	

Sample 1 Explanation

In the first test case, $7:110 \rightarrow 8:50$ by rule 1, since $110 = 60 + 50$.

In the second test case, $25:30 \rightarrow 1:30$ by rule 2, since $25 = 24 + 1$.

In the third test case, $0:0$ is already a standard time.

Problem **K** Aeonglass

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

任季节一再轮转，我却始终没能习惯。

Problem Statement

In the Euclidean plane, you are given a polygon $P = P_0P_1 \cdots P_{n-1}$ whose vertices are listed in counter-clockwise order.

An angle α ($0 \leq \alpha < \pi$) is a **stable direction** if and only if there exists a $c \in \mathbb{R}$ such that, for every point X in polygon P (including its boundary), the perpendicular segment from X to the line

$$\cos \alpha x + \sin \alpha y + c = 0$$

lies entirely in P (including its boundary). Here α is a direction of the line's normal vector.

Find the number of **stable directions**. This number can be shown to be finite.

Input

The first line contains an integer T ($1 \leq T \leq 100$), the number of test cases.

The first line of each test case contains an integer n ($3 \leq n \leq 10^5$), the number of polygon vertices.

Each of the next n lines contains two integers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$), the coordinates of the i -th vertex, listed in counterclockwise order.

Within a test case, all vertices are distinct. Adjacent edges intersect only at their shared endpoint, and nonadjacent edges do not intersect.

The sum of n over all test cases does not exceed 10^5 .

Output

For each test case, output one integer: the number of **stable directions**.

Sample 1

standard input

```
3
4
0 0
1 0
2 1
1 1
4
0 0
2 0
2 2
0 2
6
0 0
4 0
4 1
1 1
3 4
0 4
```

standard output

```
3
4
1
```

Sample 1 Explanation

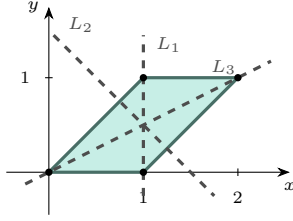


Illustration for test case 1

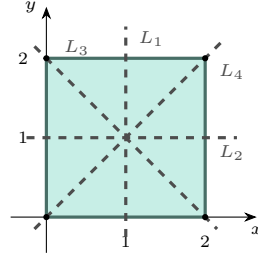


Illustration for test case 2

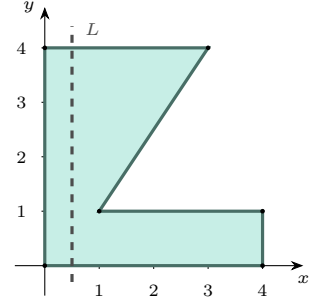


Illustration for test case 3

The lines and **stable directions** for test case 1 are

$$\begin{aligned} L_1 : \quad x - 1 &= 0, & \alpha_1 &= 0, \\ L_2 : \quad \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y - \frac{3\sqrt{2}}{4} &= 0, & \alpha_2 &= \frac{\pi}{4}, \\ L_3 : \quad -\frac{1}{\sqrt{5}}x + \frac{2}{\sqrt{5}}y &= 0, & \alpha_3 &= \pi - \arctan 2. \end{aligned}$$

The lines and **stable directions** for test case 2 are

$$\begin{aligned} L_1 : \quad x - 1 &= 0, & \alpha_1 &= 0, \\ L_2 : \quad y - 1 &= 0, & \alpha_2 &= \frac{\pi}{2}, \\ L_3 : \quad \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y - \sqrt{2} &= 0, & \alpha_3 &= \frac{\pi}{4}, \\ L_4 : \quad -\frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y &= 0, & \alpha_4 &= \frac{3\pi}{4}. \end{aligned}$$

The line and **stable direction** for test case 3 are

$$L : \quad x - \frac{1}{2} = 0, \quad \alpha_1 = 0.$$

The corresponding line is not uniquely determined.

Problem In Filtration 3

Time limit 2 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

除恶务尽——何为「恶」，我们自有裁定。

Problem Statement

Maddy is still purging Baddy's evil army...for the third time!

The game takes place on an infinite Cartesian plane, which you may imagine as an infinite chessboard.

Baddy controls n stationary **cannons**, placed at $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.

Under Chinese chess rules, a cannon attacks by jumping over a screen. If cannons A and B share a row or column and **exactly one** other cannon C lies between them, A attacks B (and B attacks A). Cannon C is their "screen."

At each step, Maddy may destroy a cannon **that is not currently attacked by any other cannon**.

The destroyed cannon is immediately removed; it can no longer attack or serve as a screen.

Maddy wants to destroy all n cannons. Does there exist an order of destruction that clears the board?

Input

The first line contains an integer T ($1 \leq T \leq 10^5$), the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 2 \times 10^5$), the number of cannons.

Each of the next n lines contains two integers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$), the position of cannon i .

The sum of n over all test cases does not exceed 2×10^5 . All cannon positions within a test case are distinct.

Output

For each test case, output **YES** if some valid destruction order clears the board, or **NO** otherwise.

The answer is case-insensitive; for example, **yes**, **YES**, **Yes**, and **YeS** are all accepted as affirmative answers.

Sample 1

standard input

```
3
3
0 0
0 1
0 2
4
0 0
1 0
2 0
3 0
8
2 0
0 1
1 1
2 1
1 2
2 2
3 2
1 3
```

standard output

```
YES
NO
NO
```

Sample 1 Explanation

In the first test case, the three cannons share a column. Initially, $(0,0)$ and $(0,2)$ attack each other over the screen at $(0,1)$, while $(0,1)$ is unattacked. Destroy $(0,1)$ first. Then the other two are unattacked and may be destroyed in either order.

In the second test case, four cannons share a row. Initially, $(0,0)$ and $(2,0)$ attack each other over $(1,0)$; $(1,0)$ and $(3,0)$ attack each other over $(2,0)$. Thus every cannon is attacked, none can be destroyed, and the answer is **NO**.

Problem M Witch Trial

Time limit 6 seconds

Memory limit 1 Gigabyte

Input standard input

Output standard output

我宁可成为异端，也不愿背叛良知。

Problem Statement

Your trident has struck a villager again. This time, when the trial ends, you discover that you can no longer find it.

You know that the trident landed on a rooted tree T with n vertices, rooted at vertex 1. Each vertex has a weight a_i , and at every moment these weights form a permutation of $1 \sim n$. For vertices u, v , let $P(u, v)$ be their simple path, including both u and v .

A set S of tree vertices is a **trident** if and only if there exist three vertices F, x, y (which may be equal) satisfying:

- F is an ancestor of x , or $F = x$;
- F is an ancestor of y , or $F = y$;
- $S = P(F, x) \cup P(F, y)$;

Process q operations in order. There are three types:

Operation 1: Lightning. Given vertices x and y , swap their current weights.

Operation 2: Judgment. Given two distinct weights $l < r$, reverse the weights within $[l, r]$.

Formally, if vertex x has weight a_x before this operation and a'_x afterward, define

$$a'_x = \begin{cases} l + r - a_x, & l \leq a_x \leq r, \\ a_x, & \text{otherwise.} \end{cases}$$

Then set $a_x \leftarrow a'_x$ simultaneously for every integer $1 \leq x \leq n$. The weights remain a permutation of $1 \sim n$.

Operation 3: Find the trident. Given $l \leq r$, determine whether $S = \{x \in N, 1 \leq x \leq n \mid l \leq a_x \leq r\}$ is a **trident**.

Answer all questions from **Operation 3** in order.

Input

The first line contains an integer T ($1 \leq T \leq 10^3$), the number of test cases.

The first line of each test case contains two integers n, q ($1 \leq n, q \leq 10^5$), the numbers of tree vertices and operations.

Each of the next $n - 1$ lines contains two integers u, v ($1 \leq u, v \leq n, u \neq v$), describing a tree edge.

The next line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$), the initial vertex weights.

Each of the next q lines describes an operation. It starts with an integer op_i ($1 \leq \text{op}_i \leq 3$), followed by:

- If $\text{op}_i = 1$, two integers x, y ($1 \leq x, y \leq n$): perform **Operation 1**, swapping their weights.
- If $\text{op}_i = 2$, two integers l, r ($1 \leq l < r \leq n$): perform **Operation 2**, reversing the weights in $[l, r]$.
- If $\text{op}_i = 3$, two integers l, r ($1 \leq l \leq r \leq n$): perform **Operation 3**, asking whether the vertices with weights in $[l, r]$ form a **trident**.

Within each test case, the initial weights form a permutation of $1 \sim n$, and the given edges form a tree with n vertices.

Across all test cases, $\sum n \leq 3 \times 10^5$ and $\sum q \leq 3 \times 10^5$.

Output

For each **Operation 3**, output one line:

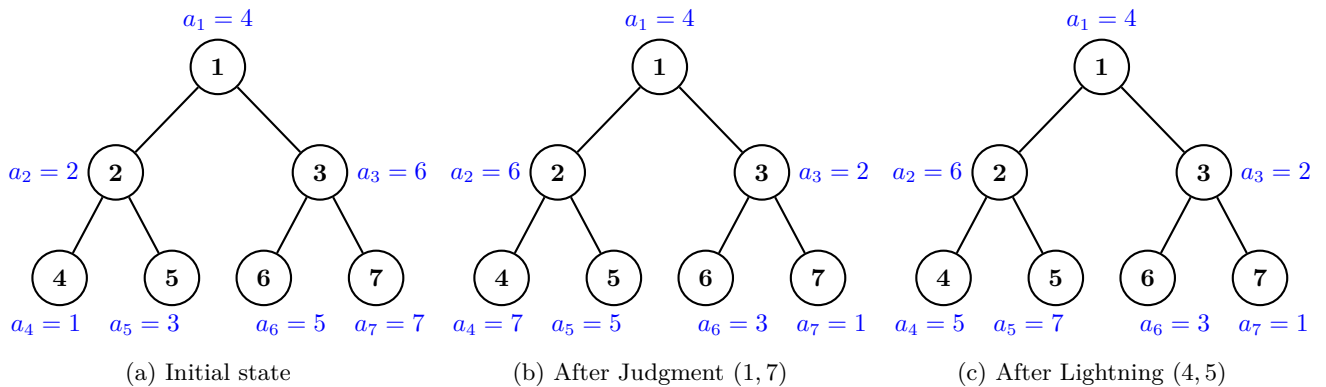
- Output **YES** if $S = \{x \in \mathbb{N}, 1 \leq x \leq n \mid l \leq a_x \leq r\}$ is a **trident**;
- Otherwise, output **NO**.

The answer is case-insensitive; for example, **yes**, **YES**, **Yes**, and **YeS** are all accepted as affirmative answers.

Sample 1

standard input	standard output
1	YES
7 6	NO
1 2	YES
1 3	YES
2 4	
2 5	
3 6	
3 7	
4 2 6 1 3 5 7	
3 1 3	
3 1 5	
2 1 7	
3 4 7	
1 4 5	
3 5 7	

Sample 1 Explanation



For the third trident query in the first test case, after the preceding Judgment operation, the current weights are

$$(a_1, a_2, \dots, a_7) = (4, 6, 2, 7, 5, 3, 1).$$

Weights 4, 5, 6, 7 are at vertices 1, 5, 2, 4, respectively, so $S = \{1, 2, 4, 5\}$. Choose $F = 1$, $x = 4$, and $y = 5$. Then:

- Vertex 1 is an ancestor of vertex 4.
- Vertex 1 is an ancestor of vertex 5.
- $P(1, 4) \cup P(1, 5) = \{1, 2, 4\} \cup \{1, 2, 5\} = S$.

Therefore, S is a **trident**.