A. Gellyfish and Flaming Peony

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   512 megabytes
```

Gellyfish hates math problems, but she has to finish her math homework:

Gellyfish is given an array of n positive integers $a_1, a_2, \ldots, a_n$.

She needs to do the following two-step operation until all elements of a are equal:

1. Select two indexes i, j satisfying $1 \leq i, j \leq n$ and $i \neq j$.
2. Replace a_i with $\gcd(a_i, a_j)$.

Now, Gellyfish asks you for the minimum number of operations to achieve her goal.

It can be proven that Gellyfish can always achieve her goal.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 5000$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 5000$) — the length of the array.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 5000$) — the elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed 5000.

Output

For each test case, output a single integer — the minimum number of operations to achieve her goal.

Standard Input	Standard Output
3 3 12 20 30 6 1 9 1 9 8 1 3 6 14 15	4 3 3

Note

In the first test case, the following is a way that minimizes the number of operations:

1. Choose $i = 3$ and $j = 2$ and replace a_3 with $\gcd(a_3, a_2) = \gcd(30, 20) = 10$, then a becomes $[12, 20, 10]$.
2. Choose $i = 1$ and $j = 3$ and replace a_1 with $\gcd(a_1, a_3) = \gcd(12, 10) = 2$, then a becomes $[2, 20, 10]$.
3. Choose $i = 2$ and $j = 1$ and replace a_2 with $\gcd(a_2, a_1) = \gcd(20, 2) = 2$, then a becomes $[2, 2, 10]$.

4. Choose $i = 3$ and $j = 1$ and replace a_3 with $\gcd(a_3, a_1) = \gcd(10, 2) = 2$, then a becomes $[2, 2, 2]$.

,

B. Gellyfish and Camellia Japonica

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   512 megabytes
```

Gellyfish has an array of n integers $c_1, c_2, \ldots, c_n$. In the beginning, $c = [a_1, a_2, \ldots, a_n]$.

Gellyfish will make q modifications to c.

For $i = 1, 2, \ldots, q$, Gellyfish is given three integers x_i, y_i, and z_i between 1 and n. Then Gellyfish will set $c_{z_i} := \min(c_{x_i}, c_{y_i})$.

After the q modifications, $c = [b_1, b_2, \ldots, b_n]$.

Now Flower knows the value of b and the value of the integers x_i, y_i, and z_i for all $1 \leq i \leq q$, but she doesn't know the value of a.

Flower wants to find any possible value of the array a or report that no such a exists.

If there are multiple possible values of the array a, you may output any of them.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and q ($1 \leq n, q \leq 3 \cdot 10^5$) — the size of the array and the number of modifications.

The second line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_i \leq 10^9$) — the value of the array c after the q modifications.

The following q lines each contain three integers x_i, y_i, and z_i ($1 \leq x_i, y_i, z_i \leq n$) — describing the i-th modification.

It is guaranteed that the sum of n and the sum of q over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, if a exists, output n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 10^9$) in a single line. Otherwise, output "-1" in a single line.

If there are multiple solutions, print any of them.

Standard Input	Standard Output
3 2 1 1 2 2 1 2 3 2 1 2 3 2 3 2 1 2 1	-1 1 2 3 1 2 3 4 5 5

6 4 1 2 2 3 4 5 5 6 6 4 5 5 3 4 4 2 3 3	

Note

In the first test case, based on the given sequence of modifications, we know that $b_1 = a_1$ and $b_2 = \min(a_1, a_2)$. Therefore, it is necessary that $b_2 \leq b_1$. However, for the given b, we have $b_1 < b_2$. Therefore, there is no solution.

In the second test case, we can see that the given c becomes b from a after the given modifications, and c is not changed at each modification.

,

C. Gellyfish and Eternal Violet

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

There are n monsters, numbered from 1 to n, in front of Gellyfish. The HP of the i-th monster is h_i.

Gellyfish doesn't want to kill them, but she wants to keep these monsters from being a threat to her. So she wants to reduce the HP of all the monsters to exactly 1.

Now, Gellyfish, with The Sword Sharpened with Tears, is going to attack the monsters for m rounds. For each round:

1. The Sword Sharpened with Tears shines with a probability of p.
2. Gellyfish can choose whether to attack:
    - If Gellyfish doesn't attack, nothing happens.
    - If Gellyfish chooses to attack and The Sword Sharpened with Tears shines, the HP of all the monsters will be reduced by 1.
    - If Gellyfish chooses to attack and The Sword Sharpened with Tears does not shine, Gellyfish can choose one of the monsters and reduce its HP by 1.

Please note that before Gellyfish decides whether or not to attack, she will know whether the sword shines or not. Also, when the sword shines, Gellyfish can only make attacks on all the monsters and cannot make an attack on only one monster.

Now, Gellyfish wants to know what the probability is that she will reach her goal if she makes choices optimally during the battle.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The first line of each test case contains three integers n, m, and p' ($1 \le n \le 20$, $1 \le m \le 4000$, $0 \le p' \le 100$) — the number of monsters, the number of rounds of attacks, and an integer representing the probability $p = \frac{p'}{100}$ that the Sword Sharpened with Tears shines.

The second line of each test case contains n integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le 400$) — the HP of the monsters.

It is guaranteed that the sum of n over all test cases does not exceed 100.

Output

For each test case, output a single real number representing the probability that Gellyfish will reach her goal.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Standard Input	Standard Output

4 2 2 10 2 2 5 5 20 2 2 2 2 2 6 20 50 1 1 4 5 1 4 9 50 33 9 9 8 2 4 4 3 5 3	0.910000 0.672320 0.588099 0.931474

Note

In the first test case, Gellyfish will always attack whether the sword shines or not in the first round.

If the sword shines in the first round, then Gellyfish can reach her goal after the attack in the first round.

Otherwise, if the sword does not shine in the first round, she will attack monster 1 in the first round. For the second round:

- If the sword shines, since monster 1 was attacked in the first round, Gellyfish can't reach her goal.
- Otherwise, Gellyfish can attack monster 2, allowing her to reach her goal.

Therefore, the probability that Gellyfish can reach her goal is $10\% + (90\% \cdot 90\%) = 91\%$.

In the second test case, Gellyfish will only attack in the first round where the sword shines. It can be observed that the only way Gellyfish can't reach her goal is if the sword never shines in all 5 rounds. The probability that Gellyfish can reach her goal is $100\% - (80\%)^5 = 67.232\%$.

,

D. Gellyfish and Forget-Me-Not

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   1024 megabytes
```

Gellyfish and Flower are playing a game.

The game consists of two arrays of n integers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$, along with a binary string $c_1 c_2 \ldots c_n$ of length n.

There is also an integer x which is initialized to 0.

The game consists of n rounds. For $i = 1, 2, \ldots, n$, the round proceeds as follows:

1. If $c_i = \texttt{0}$, Gellyfish will be the active player. Otherwise, if $c_i = \texttt{1}$, Flower will be the active player.
2. The active player will perform **exactly one** of the following operations:
    - Set $x := x \oplus a_i$.
    - Set $x := x \oplus b_i$.

Here, $\oplus$ denotes the bitwise XOR operation.

Gellyfish wants to minimize the final value of x after n rounds, while Flower wants to maximize it.

Find the final value of x after all n rounds if both players play optimally.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the number of rounds of the game.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{60}$).

The third line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($0 \le b_i < 2^{60}$).

The fourth line of each test case contains a binary string c of length n.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, output a single integer — the final value of x after all n rounds.

Standard Input	Standard Output
5 1 0 2 0 2 12 2 13 3	0 15 6 11 5

11 3 6 1 2 6 2 3 010 4 1 12 7 2 4 14 4 2 0111 9 0 5 10 6 6 2 6 2 11 7 3 15 3 6 7 6 7 8 110010010	

Note

In the first test case, there's only one round and Gellyfish is the active player of that round. Therefore, she will choose a_1, and the final value of x is 0.

In the second test case, Flower will be the active player in both rounds. She will choose a_1 and b_2, and the final value of x is $a_1 \oplus b_2 = 15$. Flower may also choose b_1 and a_2 instead for the same result of $x = a_2 \oplus b_1 = 15$.

In the third test case, $a_1 = b_1$ so it doesn't matter what decision Gellyfish makes in the first round. In the second round:

- If Flower chooses a_2, then x will become 7. Gellyfish will choose b_3 in the third round, so the final value of x will be 4.
- Otherwise, Flower chooses b_2, then x will become 4. Gellyfish will choose a_3 in the third round, so the final value of x will be 6.

Flower wants to maximize the final value of x, so Flower will choose b_2 in the second round. Therefore, the final value of x will be 6.

,

E. Gellyfish and Mayflower

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

Mayflower by Plum

May, Gellyfish's friend, loves playing a game called "Inscryption" which is played on a directed acyclic graph with n vertices and m edges. All edges $a \to b$ satisfy $a < b$.

You start in vertex 1 with some coins. You need to move from vertex 1 to the vertex where the boss is located along the directed edges, and then fight with the final boss.

Each of the n vertices of the graph contains a Trader who will sell you a card with power w_i for c_i coins. You can buy as many cards as you want from each Trader. However, you can only trade with the trader on the i-th vertex if you are currently on the i-th vertex.

In order to defeat the boss, you want the sum of the power of your cards to be as large as possible.

You will have to answer the following q queries:

- Given integers p and r. If the final boss is located at vertex p, and you have r coins in the beginning, what is the maximum sum of the power of your cards when you fight the final boss? Note that you are allowed to trade cards on vertex p.

Input

The first line of input contains two integers n and m ($1 \le n \le 200, n - 1 \le m \le \min(\frac{n(n-1)}{2}, 2000)$) — the number of vertices and the number of edges.

The i-th of the following n lines of input each contains two integers c_i and w_i ($1 \le c_i \le 200$, $1 \le w_i \le 10^9$) — describing the cards of the Trader on the i-th vertex.

In the following m lines of input, each line contains two integers u and v ($1 \le u < v \le n$), indicating a directed edge from vertex u to vertex v. It is guaranteed that every edge (u, v) appears at most once.

The next line of input contains one single integer q ($1 \le q \le 2 \cdot 10^5$) — the number of queries.

In the following q lines of input, each line contains two integers p and r ($1 \le p \le n, 1 \le r \le 10^9$).

It is guaranteed that for all i, there exists a path from vertex 1 to vertex i.

Output

For each query, output the answer to the query.

Standard Input	Standard Output
3 2 3 9 2 5 1 2 1 2 2 3	9 10 11 9 14 14

```
6
1 4
2 4
3 4
1 5
2 5
3 5
```

```
4 4
10 1000
2 5
1 2
3 9
1 2
1 3
2 4
3 4
9
2 3
3 3
4 1
4 2
4 4
4 5
4 101
4 102
4 103
```

```
5
6
2
5
11
14
10002
10005
10009
```

```
6 8
9 5
4 1
8 9
10 4
9 4
8 2
3 5
4 6
3 4
2 3
1 2
2 5
4 5
1 3
10
3 12
1 9
6 47
2 19
1 129
5 140
```

```
10
5
46
10
70
154
81
35
21
109
```

2 148 1 63 2 43 3 102	

Note

For the third query in the first example, we will play the game in the following order:

- buy 1 card with 9 power from the trader on vertex 1, and you'll still have 1 coin after the trade.
- move from vertex 1 to vertex 2.
- move from vertex 2 to vertex 3.
- buy 1 card with 2 power from the trader on vertex 3, and you'll have no coins after the trade.

In the end, we will have 1 card with 9 power and 1 card with 2, so the sum of the power of the cards is $9 + 2 = 11$.

For the fifth query in the second example, we will play the game in the following order:

- move from vertex 1 to vertex 3.
- buy 1 card with 2 power from the trader on vertex 3, and you'll still have 3 coins after the trade.
- move from vertex 3 to vertex 4.
- buy 1 card with 9 power from the trader on vertex 4, and you'll have no coins after the trade.

In the end, we will have 1 card with 2 power and 1 card with 9, so the sum of the power of the cards is $2 + 9 = 11$.

For the sixth query in the second example, we will play the game in the following order:

- move from vertex 1 to vertex 2.
- buy 1 card with 5 power from the trader on vertex 2, and you'll still have 3 coins after the trade.
- move from vertex 2 to vertex 4.
- buy 1 card with 9 power from the trader on vertex 4, and you'll have no coins after the trade.

In the end, we will have 1 card with 5 power and 1 card with 9, so the sum of the power of the cards is $5 + 9 = 14$.

For the seventh query in the second example, we will play the game in the following order:

- buy 10 cards with 1000 power from the trader on vertex 1, and you'll still have 1 coin after the trade.
- move from vertex 1 to vertex 3.
- buy 1 card with 2 power from the trader on vertex 3, and you'll have no coins after the trade.
- move from vertex 3 to vertex 4.

In the end, we will have 10 cards with 1000 power and 1 card with 2 power, so the sum of the power of the cards is $10 \cdot 1000 + 2 = 10\,002$.

,

F1. Gellyfish and Lycoris Radiata (Easy Version)

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   1024 megabytes
```

This is the easy version of the problem. The difference between the versions is that in this version, the time limit and the constraints on n and q are lower. You can hack only if you solved all versions of this problem.

Gellyfish has an array consisting of n sets. Initially, all the sets are empty.

Now Gellyfish will do q operations. Each operation contains one modification operation and one query operation, for the i-th ($1 \leq i \leq q$) operation:

First, there will be a modification operation, which is one of the following:

1. **Insert** operation: You are given an integer r. For the 1-th to r-th sets, insert element i. Note that the element inserted here is i, the index of the operation, not the index of the set.
2. **Reverse** operation: You are given an integer r. Reverse the 1-th to r-th sets.
3. **Delete** operation: You are given an integer x. Delete element x from all sets that contain x.

Followed by a query operation:

- **Query** operation: You are given an integer p. Output the smallest element in the p-th set (If the p-th set is empty, the answer is considered to be 0).

Now, Flower needs to provide the answer for each query operation. Please help her!

Additional constraint on the problem: Gellyfish will only give the next operation after Flower has answered the previous query operation. That is, you need to solve this problem **online**. Please refer to the input format for more details.

Input

The first line contains two integers n and q ($1 \leq n, q \leq 10^5$) — the number of the sets and the number of operations.

As you need to respond to the operations online, the operations will be encoded.

The i-th line of the following q lines contains three integers a, b, and c ($1 \leq a \leq 3$, $1 \leq c \leq n$) — describing the i-th operation in an encoded form.

Here, a represents the type of modification operation. Among them, $a = 1$ represents **Insert** operation, $a = 2$ represents **Reverse** operation, $a = 3$ represents **Delete** operation.

- If $a = 1$, then the modification operation is the **Insert** operation. It will be guaranteed that $1 \leq b \leq n$. r will be calculated as $r = (b + \text{ans}_{i-1} - 1) \bmod n + 1$.
- If $a = 2$, then the modification operation is the **Reverse** operation. It will be guaranteed that $1 \leq b \leq n$. r will be calculated as $r = (b + \text{ans}_{i-1} - 1) \bmod n + 1$.
- If $a = 3$, then the modification operation is the **Delete** operation. It will be guaranteed that $1 \leq b \leq q$. x will be calculated as $x = (b + \text{ans}_{i-1} - 1) \bmod q + 1$.

For the query operation, p will be calculated as $p = (c + \text{ans}_{i-1} - 1) \bmod n + 1$.

Here $\text{ans}_i (1 \leq i \leq q)$ represents the answer to the query operation in the i-th operation. Additionally, we define $\text{ans}_0 = 0$.

Output

For each query operation, output the answer to the query.

Standard Input	Standard Output
5 10 1 2 2 2 3 1 1 5 3 2 2 5 1 5 2 2 4 4 3 2 2 3 1 2 3 10 5 3 2 4	1 0 1 1 3 1 0 5 0 0

Note

All the sets are empty in the beginning, so the array is $[\{\}, \{\}, \{\}, \{\}, \{\}]$.

With the decoding method given before, we can see what happens in each operation:

1. For the first operation: $a = 1, r = 2, p = 2$. The modification operation is an **Insert** operation; element 1 is inserted into the first two sets; so the array becomes $[\{1\}, \{1\}, \{\}, \{\}, \{\}]$, and the smallest element in the second set is 1.
2. For the second operation: $a = 2, r = 4, p = 2$. The modification operation is a **Reverse** operation; the first four sets are reversed; so the array becomes $[\{\}, \{\}, \{1\}, \{1\}, \{\}]$, and the second set is empty, which means the answer is 0.
3. For the third operation: $a = 1, r = 5, p = 3$. The modification operation is an **Insert** operation; element 3 is inserted into all the sets; so the array becomes $[\{3\}, \{3\}, \{1, 3\}, \{1, 3\}, \{3\}]$, and the smallest element in the third set is 1.
4. For the fourth operation: $a = 2, r = 3, p = 1$. The modification operation is a **Reverse** operation; the first three sets are reversed; so the array becomes $[\{1, 3\}, \{3\}, \{3\}, \{1, 3\}, \{3\}]$, and the smallest element in the first set is 1.
5. For the fifth operation: $a = 1, r = 1, p = 3$. The modification operation is an **Insert** operation; element 5 is inserted into the first set; so the array becomes $[\{1, 3, 5\}, \{3\}, \{3\}, \{1, 3\}, \{3\}]$, and the smallest element in the third set is 3.
6. For the sixth operation: $a = 2, r = 2, p = 2$. The modification operation is a **Reverse** operation; the first two sets are reversed; so the array becomes $[\{3\}, \{1, 3, 5\}, \{3\}, \{1, 3\}, \{3\}]$, and the smallest element in the second set is 1.
7. For the seventh operation: $a = 3, x = 3, p = 3$. The modification operation is a **Delete** operation; element 3 is deleted from all the sets; so the array becomes $[\{\}, \{1, 5\}, \{\}, \{1\}, \{\}]$, and the third set is empty, which means the answer is 0.
8. For the eighth operation: $a = 3, x = 1, p = 2$. The modification operation is a **Delete** operation; element 1 is deleted from all the sets; so the array becomes $[\{\}, \{5\}, \{\}, \{\}, \{\}]$, and the smallest element in the second set is 5.
9. For the ninth operation: $a = 3, x = 5, p = 5$. The modification operation is a **Delete** operation; element 5 is deleted from all the sets; so the array becomes $[\{\}, \{\}, \{\}, \{\}, \{\}]$, and the fifth set is empty, which

means the answer is 0.

10. For the tenth operation: $a = 3, x = 2, p = 4$. The modification operation is a **Delete** operation; element 2 is deleted from all the sets; so the array becomes $[\{\}, \{\}, \{\}, \{\}, \{\}]$, and the fourth set is empty, which means the answer is 0.

Please note that although we have not inserted element 2 into the sets, we still delete element 2 from all the sets in the tenth operation, which means that the **Delete** operation doesn't necessarily require the existence of a set to contain the deleted element. It also shows that it is possible to have two **Delete** operations that delete the same element.

,

F2. Gellyfish and Lycoris Radiata (Hard Version)

```
Input file:     standard input
Output file:    standard output
Time limit:     5 seconds
Memory limit:   1024 megabytes
```

This is the hard version of the problem. The difference between the versions is that in this version, the time limit and the constraints on n and q are higher. You can hack only if you solved all versions of this problem.

Gellyfish has an array consisting of n sets. Initially, all the sets are empty.

Now Gellyfish will do q operations. Each operation contains one modification operation and one query operation, for the i-th ($1 \leq i \leq q$) operation:

First, there will be a modification operation, which is one of the following:

1. **Insert** operation: You are given an integer r. For the 1-th to r-th sets, insert element i. Note that the element inserted here is i, the index of the operation, not the index of the set.
2. **Reverse** operation: You are given an integer r. Reverse the 1-th to r-th sets.
3. **Delete** operation: You are given an integer x. Delete element x from all sets that contain x.

Followed by a query operation:

- **Query** operation: You are given an integer p. Output the smallest element in the p-th set (If the p-th set is empty, the answer is considered to be 0).

Now, Flower needs to provide the answer for each query operation. Please help her!

Additional constraint on the problem: Gellyfish will only give the next operation after Flower has answered the previous query operation. That is, you need to solve this problem **online**. Please refer to the input format for more details.

Input

The first line contains two integers n and q ($1 \leq n, q \leq 3 \cdot 10^5$) — the number of the sets and the number of operations.

As you need to respond to the operations online, the operations will be encoded.

The i-th line of the following q lines contains three integers a, b, and c ($1 \leq a \leq 3$, $1 \leq c \leq n$) — describing the i-th operation in an encoded form.

Here, a represents the type of modification operation. Among them, $a = 1$ represents **Insert** operation, $a = 2$ represents **Reverse** operation, $a = 3$ represents **Delete** operation.

- If $a = 1$, then the modification operation is the **Insert** operation. It will be guaranteed that $1 \leq b \leq n$. r will be calculated as $r = (b + \text{ans}_{i-1} - 1) \bmod n + 1$.
- If $a = 2$, then the modification operation is the **Reverse** operation. It will be guaranteed that $1 \leq b \leq n$. r will be calculated as $r = (b + \text{ans}_{i-1} - 1) \bmod n + 1$.
- If $a = 3$, then the modification operation is the **Delete** operation. It will be guaranteed that $1 \leq b \leq q$. x will be calculated as $x = (b + \text{ans}_{i-1} - 1) \bmod q + 1$.

For the query operation, p will be calculated as $p = (c + \text{ans}_{i-1} - 1) \bmod n + 1$.

Here $\text{ans}_i (1 \leq i \leq q)$ represents the answer to the query operation in the i-th operation. Additionally, we define $\text{ans}_0 = 0$.

Output

For each query operation, output the answer to the query.

Standard Input	Standard Output
5 10 1 2 2 2 3 1 1 5 3 2 2 5 1 5 2 2 4 4 3 2 2 3 1 2 3 10 5 3 2 4	1 0 1 1 3 1 0 5 0 0

Note

All the sets are empty in the beginning, so the array is $[\{\}, \{\}, \{\}, \{\}, \{\}]$.

With the decoding method given before, we can see what happens in each operation:

1. For the first operation: $a = 1, r = 2, p = 2$. The modification operation is an **Insert** operation; element 1 is inserted into the first two sets; so the array becomes $[\{1\}, \{1\}, \{\}, \{\}, \{\}]$, and the smallest element in the second set is 1.
2. For the second operation: $a = 2, r = 4, p = 2$. The modification operation is a **Reverse** operation; the first four sets are reversed; so the array becomes $[\{\}, \{\}, \{1\}, \{1\}, \{\}]$, and the second set is empty, which means the answer is 0.
3. For the third operation: $a = 1, r = 5, p = 3$. The modification operation is an **Insert** operation; element 3 is inserted into all the sets; so the array becomes $[\{3\}, \{3\}, \{1, 3\}, \{1, 3\}, \{3\}]$, and the smallest element in the third set is 1.
4. For the fourth operation: $a = 2, r = 3, p = 1$. The modification operation is a **Reverse** operation; the first three sets are reversed; so the array becomes $[\{1, 3\}, \{3\}, \{3\}, \{1, 3\}, \{3\}]$, and the smallest element in the first set is 1.
5. For the fifth operation: $a = 1, r = 1, p = 3$. The modification operation is an **Insert** operation; element 5 is inserted into the first set; so the array becomes $[\{1, 3, 5\}, \{3\}, \{3\}, \{1, 3\}, \{3\}]$, and the smallest element in the third set is 3.
6. For the sixth operation: $a = 2, r = 2, p = 2$. The modification operation is a **Reverse** operation; the first two sets are reversed; so the array becomes $[\{3\}, \{1, 3, 5\}, \{3\}, \{1, 3\}, \{3\}]$, and the smallest element in the second set is 1.
7. For the seventh operation: $a = 3, x = 3, p = 3$. The modification operation is a **Delete** operation; element 3 is deleted from all the sets; so the array becomes $[\{\}, \{1, 5\}, \{\}, \{1\}, \{\}]$, and the third set is empty, which means the answer is 0.
8. For the eighth operation: $a = 3, x = 1, p = 2$. The modification operation is a **Delete** operation; element 1 is deleted from all the sets; so the array becomes $[\{\}, \{5\}, \{\}, \{\}, \{\}]$, and the smallest element in the second set is 5.
9. For the ninth operation: $a = 3, x = 5, p = 5$. The modification operation is a **Delete** operation; element 5 is deleted from all the sets; so the array becomes $[\{\}, \{\}, \{\}, \{\}, \{\}]$, and the fifth set is empty, which

means the answer is 0.

10. For the tenth operation: $a = 3, x = 2, p = 4$. The modification operation is a **Delete** operation; element 2 is deleted from all the sets; so the array becomes $[\{\}, \{\}, \{\}, \{\}, \{\}]$, and the fourth set is empty, which means the answer is 0.

Please note that although we have not inserted element 2 into the sets, we still delete element 2 from all the sets in the tenth operation, which means that the **Delete** operation doesn't necessarily require the existence of a set to contain the deleted element. It also shows that it is possible to have two **Delete** operations that delete the same element.