

# Козак Вус та граф

Назва вхідного файлу:	<code>standard input</code>
Назва вихідного файлу:	<code>standard output</code>
Ліміт часу:	6 seconds
Ліміт використання пам'яті:	1024 megabytes

Дано граф з  $n$  вершин. Також дано  $q$  запитів трьох типів:

1. У компоненті вершини  $v_i$  потрібно знайти номер  $k_i$ -ої найменшої за номером вершини. Якщо такої немає, то потрібно повернути  $-1$ . Компонента вершини  $v_i$  — це множина усіх вершин, до яких можна дістатися з вершини  $v_i$  по ребрах.
2. Додати до графа ребро, що з'єднує вершини  $u_i$  та  $v_i$ .
3. Повернутися до стану, який був після виконання  $x_i$  операцій.

Знайдіть відповіді на всі запити першого типу.

## Формат вхідних даних

Перший рядок містить три цілі числа  $n, q, g$  ( $1 \leq n, q \leq 5 \cdot 10^5, 0 \leq g \leq 9$ ).

Кожен з наступних рядків описує запит.

1.  $v_i, k_i$  ( $1 \leq v_i, k_i \leq n$ ).
2.  $v_i, u_i$  ( $1 \leq v_i, u_i \leq n$ ).
3.  $x_i$  ( $0 \leq x_i < i$ ).

## Формат вихідних даних

Для кожного запиту першого типу виведіть відповідь на запит.

## Система оцінки

1. (6 балів):  $n, q \leq 100$ ; немає операцій другого та третього типів.
2. (7 балів):  $n, q \leq 100$ ; немає операцій третього типу.
3. (4 бали):  $n, q \leq 100$ .
4. (9 балів):  $n, q \leq 3 \cdot 10^5$ ; гарантується, що в операціях другого типу  $|v_i - u_i| = 1$ ; немає запитів третього типу.
5. (8 балів):  $n, q \leq 3 \cdot 10^5$ , немає запитів третього типу.
6. (10 балів):  $n, q \leq 3 \cdot 10^5$ ; гарантується, що в операціях другого типу  $|v_i - u_i| = 1$ .
7. (19 балів):  $n, q \leq 10^5$ .
8. (17 балів):  $n, q \leq 3 \cdot 10^5$ .
9. (20 балів): без додаткових обмежень.

## Приклади

standard input	standard output
10 12 0 1 1 1 2 1 2 2 1 6 2 9 10 2 3 10 2 10 6 1 1 5 1 1 3 3 5 1 1 3 3 0 1 1 3	1 9 3 6 -1
10 17 0 2 1 2 1 2 2 2 3 4 1 3 2 2 6 7 2 7 8 1 7 2 1 7 3 2 5 6 1 5 5 1 5 4 2 5 4 2 3 2 1 1 7 1 1 4 1 1 8 1 1 9	2 4 7 8 -1 8 7 4 8 -1
6 14 0 2 1 6 2 1 3 1 3 2 1 3 3 1 1 1 1 2 1 1 2 6 2 1 2 2 2 2 2 2 1 2 1 5 2 1 4 1 6 6 1 1 5	3 6 1 2 -1 6 5
5 5 0 2 1 2 1 1 2 3 0 2 1 3 1 1 2	2 3