

Задача А. СНМ

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ваша задача — реализовать **Persistent Disjoint-Set-Union**. Что это значит?

Про **Disjoint-Set-Union**:

Изначально у вас есть n элементов. Нужно научиться отвечать на 2 типа запросов.

- $+ a b$ — объединить множества, в которых лежат вершины a и b
- $? a b$ — сказать, лежат ли вершины a и b сейчас в одном множестве

Про **Persistent**:

Теперь у нас будет несколько копий (версий) структуры данных **Disjoint-Set-Union**.

Запросы будут выглядеть так:

- $+ i a b$ — запрос к i -й структуре, объединить множества, в которых лежат вершины a и b . При этом i -я структура остается не измененной, создается новая версия, ей присваивается новый номер (какой? читайте дальше)
- $? i a b$ — запрос к i -й структуре, сказать, лежат ли вершины a и b сейчас в одном множестве

Формат входных данных

На первой строке 2 числа N ($1 \leq N \leq 10^5$) и K ($0 \leq K \leq 10^5$) — число элементов и число запросов. Изначально все элементы находятся в различных множествах. Эта начальная копия (версия) структуры имеет номер 0.

Далее следуют K строк, на каждой описание очередного запроса. Формат запросов описан выше. Запросы нумеруются целыми числами от 1 до K .

Пусть j -й из K запросов имеет вид « $+ i a b$ ». Тогда новая версия получит номер j . Запросы вида « $? i a b$ » не порождают новых структур.

Формат выходных данных

Для каждого запроса вида $? i a b$ на отдельной строке нужно вывести YES или NO.

Пример

стандартный ввод	стандартный вывод
4 7	NO
+ 0 1 2	YES
? 0 1 2	YES
? 1 1 2	YES
+ 1 2 3	NO
? 4 3 1	
? 0 4 4	
? 4 1 4	