

SBSWAP: Subtree swapping

Легенда, описанная далее, переформулирована и упрощена переводчиком, чтобы читатель мог лучше понять условие задачи. Оригинальную легенду вы можете прочитать на странице задачи в контексте.

Условие:

Дано корневое вершинно-взвешенное дерево, имеющее N вершин, пронумерованных от 1 до N . Вершина под номером 1 является корнем дерева. Каждой вершине дерева сопоставлено целое число. Ваша задача - обработать Q запросов следующих трех типов:

- **getSum(u)** - вернуть сумму значений вершин поддерева с корнем u .
- **add(u, x)** - добавить x к значению каждой вершины из поддерева с корнем u .
- **swap(u, v)** - поменять местами поддерева с корнями u и v , только если они не пересекаются.

Формат ввода:

Первая строка содержит разделенные пробелами целые числа N и Q - количество вершин в дереве и количество запросов соответственно.

Вторая строка содержит N разделенных пробелами целых чисел w_1, w_2, \dots, w_N , где w_i обозначает значение, сопоставленное вершине i .

Каждая из следующих $N-1$ строк содержит два разделенных пробелами целых числа u и v - описание ребра между вершинами под номерами u и v .

Каждая из следующих Q строк содержит описание запроса. В каждой строке первое целое число **qtype** обозначает тип запроса.

- Если **qtype** = **1**, то строка содержит еще одно целое число u .
- Если **qtype** = **2** или **qtype** = **3**, то строка содержит еще пару разделенных пробелом целых чисел - параметры запроса.

Формат вывода:

Для каждого запроса **первого** типа выведите в отдельную строку единственное целое число - ответ на запрос. Более того, для каждого запроса **третьего** типа выведите **-1**, если поддерева пересекаются.

Ограничения:

- $1 \leq w_i \leq 10^5$

Подзадачи:

- Подзадача 1 (10 баллов): $1 \leq N, Q \leq 1000$
- Подзадача 2 (20 баллов): $1 \leq N, Q \leq 10^5$, нет запросов третьего типа.
- Подзадача 3 (70 баллов): $1 \leq N, Q \leq 10^5$

Примеры тестов:

Входные данные:

```
10 5
1 1 1 1 1 1 1 1 1 1
1 2
1 3
1 8
3 4
```

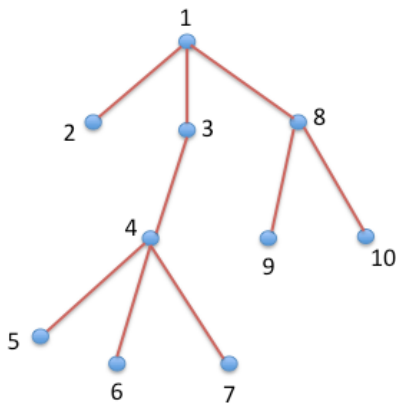
```
8 9
8 10
4 5
4 6
4 7
2 8 1
1 3
3 4 8
1 3
3 1 2
```

Выходные данные:

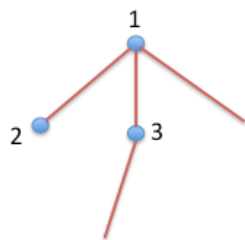
```
5
7
-1
```

Пояснения:

Изначально дерево имеет **10** вершин, каждой из которых сопоставлено целое число 1. Этот граф изображен на рисунке:

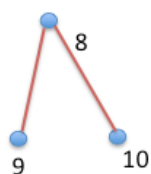
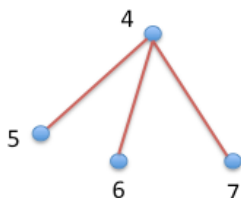


Дано 5 запросов. Первый запрос добавляет 1 ко всем вершинам поддерева с корнем 8. Второй запрос требует найти сумму всех значений вершин поддерева с корнем 3, т.е. значение вершине 5, которое изначально равно 1. Третий запрос меняет местами поддерева вершин 4 и 8. Выполнение последнего запроса изображено на рисунке.

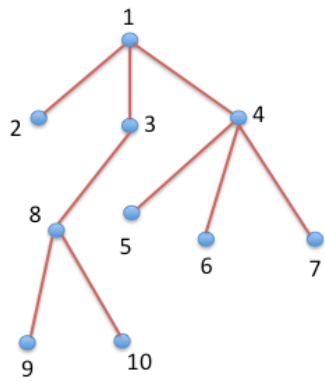


Отрезать поддерево с корнем 4

Отрезать поддерево с корнем 8



После обмена этих поддеревьев



Этот обмен может быть выполнен, т.к. поддеревья не пересекаются. Четвертый запрос требует найти сумму всех значений вершин поддерева с корнем 3. Искомая сумма равна 7, т.к. поддерево состоит из вершин со значениями 1, 2, 2, 2. Последний запрос пытается поменять местами поддеревья с корнями 1 и 2, но это сделать невозможно, т.к. поддеревья пересекаются. Следовательно, будет выведена -1.