

Задача А. Кто прочитал тот сдохнет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Однажды вам приснился кошмар. Вы проснулись и записали условие задачи из сна: Вам дано подвешенное дерево на n вершинах, у каждой вершины v есть свой целый вес w_v . Далее вам поступают m запросов одного из следующих типов:

- $0\ x\ y$ – присвоить всем вершинам в поддереве вершины x вес y .
- $1\ x$ – поменять корень на вершину x .
- $2\ x\ y\ z$ – присвоить всем вершинам на пути от вершины x до вершины y вес z .
- $3\ x$ – вывести минимум в поддереве вершины x .
- $4\ x$ – вывести максимум в поддереве вершины x .
- $5\ x\ y$ – увеличить вес всех вершин в поддереве вершины x на y .
- $6\ x\ y\ z$ – увеличить вес всех вершин на пути от x до y на z .
- $7\ x\ y$ – вывести минимум на пути от вершины x до вершины y .
- $8\ x\ y$ – вывести максимум на пути от вершины x до вершины y .
- $9\ x\ y$ – изменить предка вершины x на вершину y , если y находится в поддереве вершины x ничего не делать не надо.
- $10\ x\ y$ – вывести сумму на пути от x до y .
- $11\ x$ – вывести сумму в поддереве вершины x .

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 10^5$) — число вершин и число запросов соответственно.

В следующих $n - 1$ строках содержатся по два целых числа, описывающих ребра.

В следующих n строках содержатся изначальные веса вершин.

Следующая строка содержит корень дерева.

Следующие m строк содержат запросы в описанном выше формате. Все вершины даны в 1-нумерации. Гарантируется, что все промежуточные вычисления помещаются в `int32_t`.

Формат выходных данных

Выведите ответы на все запросы, по одному числу в строчке.

Примеры

стандартный ввод	стандартный вывод
5 5	9
2 1	1
3 1	1
4 1	
5 2	
4	
1	
4	
1	
2	
1	
10 2 3	
3 1	
7 3 4	
6 3 3 2	
9 5 1	
10 12	173
2 1	860
3 2	103
4 2	791
5 3	608
6 4	1557
7 5	
8 2	
9 4	
10 9	
791	
868	
505	
658	
860	
623	
393	
717	
410	
173	
4	
0 8 800	
1 4	
2 8 2 103	
3 9	
4 4	
5 7 304	
6 8 8 410	
7 10 8	
8 1 8	
9 6 9	
10 2 3	
11 5	

Задача В. Можно и так

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево из n вершин с корнем в вершине 1. В каждой вершине записано значение a_v . Определим красоту вершины следующим образом:

- 1) Если вершина лист, то ее красота равна a_v .
- 2) Если вершина не лист, то ее красота равна $a_v +$ произведение красот детей.

Красота дерева определяется как красота корня. Иногда значения в вершинах меняются, и вам нужно научиться пересчитывать красоту дерева.

Формат входных данных

В первой строке даны два числа n, q — число вершин в дереве и запросов ($2 \leq n, q \leq 2 \cdot 10^5$).

Во второй строке дан массив целых чисел p_2, p_3, \dots, p_n — предки вершин в дереве ($1 \leq p_i < i$).

В третьей строке дан массив целых чисел a_1, a_2, \dots, a_n — значения в вершинах ($0 \leq a_i < 998244353$).

В каждой из следующих строках вводятся запросы в формате v_i, x_i — x_i новое значение вершины v_i ($1 \leq v_i \leq n, 0 < x_i < 998244353$).

Формат выходных данных

Для каждого запроса выведите новую красоту дерева.

Примеры

стандартный ввод	стандартный вывод
3 2 1 1 3 4 1 3 4 2 1	19 7
5 4 1 1 2 2 2 5 3 3 1 3 3 5 0 4 5 5 2	26 17 17 47

Задача С. Освещение дорог

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Берляндия — страна с не очень развитой дорожной системой. Всего в Берляндии есть n городов и $n - 1$ двухсторонняя дорога, i -я из которых соединяет города v_i и u_i . Известно, что между каждой парой городов можно добраться, используя только данные дороги.

Сейчас в Берляндии ночь и на некоторых дорогах надо включить освещение. Есть закон, созданный в целях экономии электричества, запрещающий включать освещение одновременно на двух дорогах, имеющих общий конец. Жителям Берляндии интересно максимальное количество дорог, на которых можно включить освещение так, чтобы не нарушить закон, поэтому они обратились к вам за помощью.

К сожалению, в Берляндии на некоторых дорогах может идти сильная метель, из-за чего связность некоторых пар городов может нарушиться. Изначально ни на какой дороге не идёт метель. Поступает q запросов двух типов:

1. Изменить погоду на дороге e_i ($1 \leq e_i \leq n - 1$): если на дороге e_i сейчас не идёт метель, то метель начинается, и наоборот.
2. Требуется включить освещение на максимальном количестве дорог, таких что до их обоих концов можно добраться от города x_i , используя только дороги, на которых не идёт метель. Разумеется, освещение можно включать не нарушая закон, то есть не должно быть пары дорог со включенным освещением, выходящих из одного города. Другими словами, надо решить исходную задачу, если оставить только города, достижимые из x_i по дорогам, на которых не идёт метель.

Формат входных данных

Первая строка содержит единственное целое число g ($0 \leq g \leq 7$) — номер группы тестов.

Вторая строка содержит единственное целое число n ($2 \leq n \leq 300\,000$) — количество городов.

Следующие $n - 1$ строки описывают дороги. В i -й из них содержатся два целых числа v_i и u_i ($1 \leq v_i, u_i \leq n$) — номера городов, соединённых i -й дорогой. Гарантируется, что между каждой парой городов можно добраться, используя только данные дороги.

Следующая строка содержит единственное целое число q ($1 \leq q \leq 300\,000$) — количество запросов.

В следующих q строках заданы запросы. В i -й из них в начале содержится целое число t_i ($1 \leq t_i \leq 2$).

- Если $t_i = 1$, то это запрос первого типа, далее строка содержит единственное целое число e_i ($1 \leq e_i \leq n - 1$) — номер дороги, на которой меняется погода.
- Если $t_i = 2$, то это запрос второго типа, далее строка содержит единственное целое число x_i ($1 \leq x_i \leq n$). В этом случае надо решить исходную задачу, если оставить только города, достижимые из x_i по дорогам, на которых не идёт метель.

Формат выходных данных

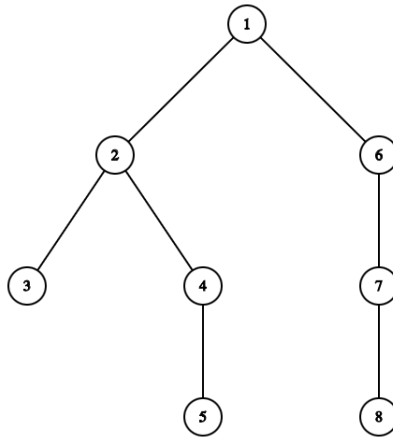
Для каждого запроса второго типа выведите максимальное число дорог, соединяющих города, достижимые из x_i , на которых можно включить освещение, не нарушая закон.

Пример

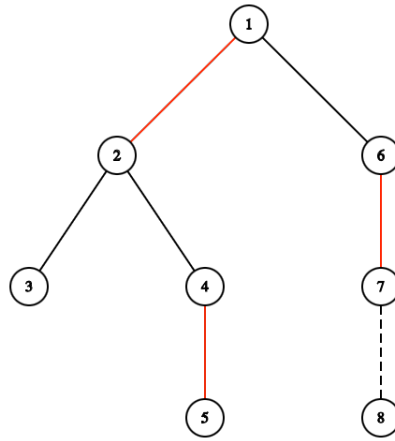
стандартный ввод	стандартный вывод
0	3
8	4
1 2	3
2 3	1
2 4	
4 5	
1 6	
6 7	
7 8	
8	
1 7	
2 1	
1 7	
2 1	
1 3	
2 3	
1 5	
2 6	

Замечание

В тестовом примере Берляндия изначально имеет следующий вид:

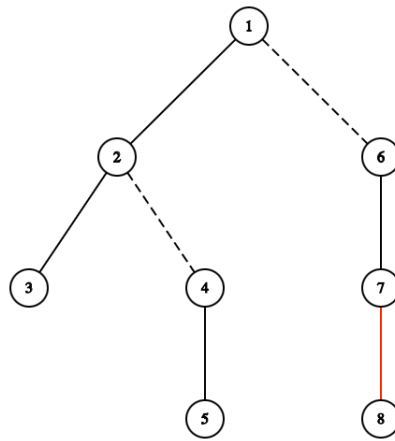


После первого запроса на дороге между городами 7 и 8 начинает идти метель. Далее приходит запрос про город 1. В этом случае мы рассматриваем все города, кроме 8-го, так как он не достижим из города 1. Ниже приведен один из оптимальных способов включить освещение на дорогах (красным отмечены дороги, на которых включено освещение):



Следующий запрос означает, что на дороге между городами 7 и 8 метель прекращается, то есть все возвращается в исходное положение.

В самом последнем запросе из вершины 6 достижимы только вершины 6, 7 и 8, поэтому можно включить освещение максимум на одной дороге, например:



Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	q		
0	0	–	–	–	Тесты из условия.
1	14	$n \leq 100$	$q \leq 100$	0	
2	13	$n \leq 100\,000$	$q \leq 100\,000$	–	$v_i = i, u_i = i + 1$
3	10	$n \leq 100\,000$	$q = 1$	–	
4	12	$n \leq 100\,000$	$q \leq 100\,000$	–	$v_i = i + 1, u_i = \lfloor \frac{v_i}{2} \rfloor$
5	19	$n \leq 100\,000$	$q \leq 100\,000$	3	Метель не может закончиться
6	20	$n \leq 100\,000$	$q \leq 100\,000$	0 – 5	
7	12	–	–	0 – 6	Offline-проверка.