

## Задача А. Доминирующие индексы

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Задано корневое неориентированное дерево, состоящее из  $n$  вершин. Вершина 1 является корнем. Определим массив глубин вершины  $x$ , как бесконечную последовательность  $[d_{x,0}, d_{x,1}, d_{x,2}, \dots]$ , где  $d_{x,i}$  — это количество вершин  $y$  таких, что выполняются оба следующих условия:

- $x$  является предком  $y$
- простой путь из  $x$  в  $y$  проходит ровно по  $i$  ребрам.

Доминирующий индекс массива глубин вершины  $x$  (или же просто доминирующий индекс вершины  $x$ ) — это такой индекс  $j$ , что:

- для каждого  $k < j, d_{x,k} < d_{x,j}$
- для каждого  $k > j, d_{x,k} \leq d_{x,j}$

Для каждой вершины дерева вычислите ее доминирующий индекс.

### Формат входных данных

В первой строке записано одно целое число  $n$  ( $1 \leq n \leq 10^6$ ) — количество вершин в дереве.

Затем следует  $n - 1$  строка, в каждой записаны по два целых числа  $x, y$  ( $1 \leq x, y \leq n, x \neq y$ ) — описание ребер дерева.

### Формат выходных данных

Выведите  $n$  чисел.  $i$ -е число должно быть равно доминирующему индексу вершины  $i$ .

### Примеры

стандартный ввод	стандартный вывод
4	0
1 2	0
2 3	0
3 4	0
4	1
1 2	0
1 3	0
1 4	0
4	2
1 2	1
2 3	0
2 4	0

## Задача В. Персистентная приоритетная очередь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Требуется реализовать структуру данных, которая хранит мультимножество и умеет изменять любую свою предыдущую версию, выполняя одну из этих операций:

1. Заданы  $v$  и  $x$ , требуется добавить в множество  $v$  элемент со значением  $x$ , после чего вывести минимальный элемент в получившемся множестве.
2. Заданы  $v$  и  $u$ , требуется объединить множества с номерами  $v$  и  $u$ , после чего вывести минимальный элемент в получившемся множестве.
3. Задано  $v$ , требуется вывести минимальный элемент в множестве  $v$ , после чего удалить минимальный элемент из множества  $v$ . Если множество пустое, то вывести, что множество пустое, и создать новое пустое множество.

Изначально есть одно пустое множество с номером 0. После операции с номером  $i$  множество, получаемое во время этой операции, получает номер  $i$ .

### Формат входных данных

Первая строка содержит число  $n$  — количество операций для выполнения.

От вас потребуется отвечать на запросы в онлайн, при этом поддерживая переменную  $s$ . Она изначально равна нулю. После каждой операции, она пересчитывается следующим образом через предыдущее значение: если ответ на запрос равен  $x$ , то  $s = (s_{old} + x) \bmod 239017$ . Если же ответом на запрос является слово `empty`, то  $s$  не изменяется.

В следующих  $n$  строках заданы запросы.

Запросы первого типа описываются строкой  $1\ a\ b$ , где  $a$  и  $b$  — неотрицательные целые числа, которые описывают  $v$  и  $x$  для соответствующего запроса, как  $v = (a + s) \bmod i$  и  $x = (b + 17s) \bmod (10^9 + 1)$ , где  $i$  — номер соответствующего запроса.

Запросы второго типа описываются строкой  $2\ a\ b$ , где  $a$  и  $b$  — неотрицательные целые числа, которые описывают  $v$  и  $u$  для соответствующего запроса, как  $v = (a + s) \bmod i$  и  $u = (b + 13s) \bmod i$ , где  $i$  — номер соответствующего запроса.

Запросы третьего типа описываются строкой  $3\ a$ , где  $a$  — неотрицательное целое число, которые описывает  $v$  для соответствующего запроса, как  $v = (a + s) \bmod i$ , где  $i$  — номер соответствующего запроса.

Число запросов не превышает 200 000. Гарантируется, что мощность любого созданного мультимножества не превышает  $2^{63}$ .

### Формат выходных данных

Требуется вывести ровно  $n$  строк, в каждой строке должно находиться неотрицательное целое число либо слово `empty`.

Для запросов первого и второго типа требуется вывести значение минимального элемента в только что созданном множестве, либо слово `empty`, если множество пустое.

Для запросов третьего типа требуется вывести минимальный элемент в множестве, либо слово `empty`, если множество пустое.

## Примеры

стандартный ввод	стандартный вывод
9	2
1 0 2	3
1 0 999999970	2
2 2 0	2
3 0	2
2 4 4	2
3 0	2
3 0	3
3 0	empty
3 8	

## Задача С. Фибоначчиева куча

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 1024 мегабайта

Эта задача для тех, кто завалил регион (или его дисквалифицировали), отчаялся в олимпиадной проге и решил полностью посвятить себя теоретической информатике и продвинутым алгоритмам. Вам предстоит написать самую настоящую Фибоначчиеву кучу. Все запросы будут задаваться массивом  $a_i$ , где  $a_1$  вам дано изначально, а для  $i > 1$   $a_i = (a_{i-1} * b + c) \bmod 2^{32}$ .

К вам будут поступать запросы двух типов:

1. В случае, если  $a_i$  чётно, то в  $i$ -м запросе от вас будет требоваться добавить число  $a_i$  в кучу.
2. В случае, если  $a_i$  нечётно от вас будет требоваться узнать значение верхнего (максимального) элемента в куче.

### Формат входных данных

В первой строке вам дано единственное число  $n$  ( $1 \leq n \leq 10^8$ ). В следующей строке вам дано 3 числа  $a_1$   $b$  и  $c$  ( $0 \leq a_1, b, c \leq 2^{32}$ ). Гарантируется, что  $a_1$  чётно.

### Формат выходных данных

Для всех запросов второго типа выведите сумму ответов на них.

### Пример

стандартный ввод	стандартный вывод
10 2 2 1	18

### Замечание

Так как Фибоначчиева куча сложная и её код трудно читать, в этой задаче не будет ревью кода.

## Задача D. Биномиальная куча

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Реализуйте биномиальную кучу.

### Формат входных данных

В первой строке содержится два целых числа:  $N$  — общее количество куч и  $M$  — количество операций ( $1 \leq N \leq 1000, 1 \leq M \leq 1\,000\,000$ ). Изначально все кучи пусты.

Требуется поддерживать следующие операции:

- 0  $a$   $v$  — добавить элемент со значением  $v$  в кучу с номером  $a$ . Вновь добавленный элемент имеет уникальный индекс равный порядковому номеру соответствующей операции добавления. Нумерация начинается с единицы.
- 1  $a$   $b$  — переложить все элементы из кучи с номером  $a$  в кучу с номером  $b$ . После этой операции куча  $a$  становится пустой.
- 2  $i$  — удалить элемент с индексом  $i$ .
- 3  $i$   $v$  — присвоить элементу с индексом  $i$  значение  $v$ . Гарантируется, что элемент существует.
- 4  $a$  — вывести на отдельной строке значение минимального элемента в куче с номером  $a$ . Гарантируется, что куча не пуста.
- 5  $a$  — удалить минимальный элемент из кучи с номером  $a$ . Если таковых несколько, то выбирается элемент с минимальным индексом. Гарантируется, что куча не пуста.

### Формат выходных данных

Для каждой операции поиска минимального элемента выведите единственное число: значение искомого элемента.

### Пример

стандартный ввод	стандартный вывод
3 19	10
0 1 10	5
4 1	7
0 2 5	7
0 2 7	10
4 2	3
3 2 20	10
4 2	8
1 2 1	
4 1	
5 1	
4 1	
3 2 3	
4 1	
2 2	
4 1	
0 1 9	
1 1 3	
0 3 8	
4 3	

## Задача Е. Повышение сотрудников

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

В подразделении компании работает  $n$  сотрудников, занумерованных числами от 1 до  $n$ . Руководитель подразделения имеет номер 1. У каждого сотрудника, кроме руководителя подразделения, есть ровно один непосредственный руководитель из числа сотрудников подразделения. Каждый сотрудник обладает единственной числовой характеристикой — компетентностью, для  $i$ -го сотрудника она равна  $c_i$ .

В ближайшее время планируется провести  $k$  переводов из подразделения в центральный офис. Каждый перевод происходит так: текущий руководитель подразделения переходит в центральный офис, его место занимает один из непосредственных подчиненных. В свою очередь, на место повышенного подчиненного назначается один из его непосредственных подчиненных и так далее.

HR-менеджер хотел бы определить, какую максимальную суммарную компетентность могут иметь  $k$  сотрудников, переведенных из подразделения в центральный офис.

### Формат входных данных

В первой строке даны два числа  $n, k$  ( $1 \leq k \leq n \leq 5 \cdot 10^3$ ) — количество сотрудников в подразделении и количество переводов в центральный офис соответственно.

Во второй строке содержатся  $n$  целых чисел  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq 10^9$ ) — компетентность сотрудников.

В третьей строке задано  $n - 1$  число  $p_2, p_3, \dots, p_n$  ( $1 \leq p_i < i$ ). Число  $p_i$  означает, что сотрудник с номером  $i$  является непосредственным подчиненным сотрудника с номером  $p_i$ . Обратите внимание, что нумерация начинается с 2, т.к. сотрудник 1 не имеет непосредственного руководителя.

### Формат выходных данных

Выведите единственное число — максимальную суммарную компетентность сотрудников, переведенных в центральный офис.

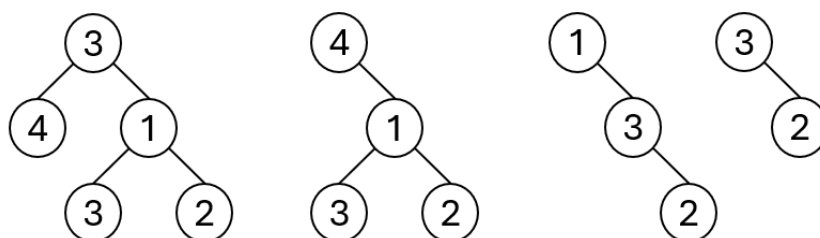
### Примеры

стандартный ввод	стандартный вывод
3 2 1 2 3 1 1	4
5 3 3 4 1 3 2 1 1 3 3	8

### Замечание

В первом примере у руководителя два непосредственных подчиненных с номером 2 и 3 и компетентностью 2 и 3 соответственно. При первом переходе к суммарной компетентности прибавляется 1 (компетентность руководителя) место руководителя занимает подчиненный с компетентностью 3. При следующем переходе к суммарной компетентности прибавляется 3, а единственным оставшимся сотрудником и новым руководителем подразделения оказывается сотрудник с компетентностью 2.

Изменение структуры подразделения во втором примере показано на рисунке:



## Задача F. Отсортируй отрезки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 512 мегабайт

Дан массив  $a$ , состоящий из  $n$  целых чисел. Поступает  $q$  запросов следующего вида:

- В ответ на  $i$ -й запрос решение должно выводить «YES», если отрезок  $[l_i, r_i]$  массива отсортирован и «NO» в ином случае.
- После ответа на  $i$ -й запрос требуется отсортировать отрезок с заданными границами.

От участника требуется быстро обработать все запросы.

### Формат входных данных

В первой строке содержится пара целых чисел  $n, q$  ( $1 \leq n \leq 3 \cdot 10^5, 1 \leq q \leq 3 \cdot 10^5$ ) — количество элементов массива и количество запросов.

Следующая строка содержит  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — элементы массива.

Следующие  $q$  строк содержат пары целых чисел  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq n$ ) — описание запросов.

### Формат выходных данных

Выходной файл должен содержать  $q$  строк, в каждой из которых записано «NO» или «YES» в зависимости от ответа на  $i$ -й запрос.

### Примеры

стандартный ввод	стандартный вывод
5 3 5 2 3 4 1 1 3 3 4 1 4	NO NO YES
7 4 3 6 2 1 2 5 7 2 5 1 4 5 7 3 6	NO NO NO YES