Общая информация по задачам второго тура

Доступ к результатам проверки решений задач во время тура

Во всех задачах вы можете неограниченное число раз запрашивать результат окончательной проверки. Для каждой подзадачи сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте.

Ограничение на размер исходного кода программы-решения

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

Процесс тестирования

Во всех задачах баллы за подзадачу начисляются только при прохождении всех тестов подзадачи. В таблице системы оценивания для каждой подзадачи указаны номера *необходимых подзадач*. Тесты для данной подзадачи запускаются только, если все необходимые подзадачи пройдены.

7. Курьерская служба	2 секунды	512 МБ

Замечания

В задачах с большим размером входных данных жюри рекомендует участникам, использующим языки группы `C/C++`, отправлять решения на проверку с использованием компилятора `LINUX GNU C++` и считывать данные, используя `scanf`.

Обратите внимание, что на этом туре доступна возможность отправки решений на Java и Python, а также решений интерактивной задачи под операционной системой Ubuntu Linux.

Задача 7. Курьерская служба

Имя входного файла:	`twopaths.in`
Имя выходного файла:	`twopaths.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В корпорации работают n сотрудников, один из которых является директором. У каждого сотрудника, кроме директора, имеется ровно один непосредственный руководитель.

Каждый сотрудник выполняет свою строго определённую работу. Если сотруднику a требуется выполнить работу, которой занимается сотрудник b, то он должен отправить сотруднику b заявку. Согласно корпоративной политике напрямую заявки могут передаваться только либо от сотрудника его непосредственному руководителю, либо наоборот — от руководителя к непосредственному подчинённому. Заявка передается от сотрудника к сотруднику до тех пор, пока не достигнет сотрудника b.

Чтобы разгрузить сотрудников, корпорация наняла k курьеров. Курьер с номером i закреплён за парой сотрудников a_i и b_i и занимается доставкой заявок между ними. Если одному сотруднику из этой пары требуется передать заявку другому, то он отдает заявку курьеру, который последовательно передает заявку между сотрудниками в соответствии с корпоративной политикой, пока заявка не пройдет всех необходимых промежуточных сотрудников и окажется доставленной адресату. В процессе доставки одной заявки курьер не посещает одного и того же сотрудника более одного раза.

С целью оптимизации расходов решено найти пару курьеров, пути следования которых перекрываются наиболее сильно, и отказаться от услуг одного из них, поручив его работу второму. Назовём *степенью дублирования* для пары курьеров количество переходов между сотрудником и его непосредственным начальником в любом направлении, входящих в путь как первого, так и второго курьера.

Требуется написать программу, которая определит двух курьеров с наибольшей степенью дублирования.

Формат входных данных

Первая строка входных данных содержит два целых числа: n — количество сотрудников и k — количество курьеров ($2 \leqslant n, k \leqslant 2 \cdot 10^5$). Сотрудники пронумерованы от 1 до n, директор имеет номер 1.

Вторая строка содержит $(n-1)$ целых чисел: $p_2, p_3, \ldots, p_n$ — номера непосредственных руководителей каждого сотрудника, кроме директора ($1 \leqslant p_i < i$).

Следующие k строк содержат по два целых числа: a_i и b_i — номера сотрудников в паре, за которой закреплён курьер с номером i ($1 \leqslant a_i, b_i \leqslant n$; $a_i \neq b_i$). За одной и той же парой сотрудников может быть закреплено несколько курьеров.

Формат выходных данных

Первая строка выходных данных должна содержать единственное целое число — наибольшую степень дублирования двух курьеров.

Вторая строка выходных данных должна содержать два различных целых числа от 1 до k — номера курьеров, входящих в пару с наибольшей степенью дублирования. Если таких пар несколько, можно вывести любую из них.

Замечание

Будем говорить, что сотрудник a *подчиняется приказам* сотрудника b, если b является непосредственным начальником a, или же непосредственный начальник a подчиняется приказам сотрудника b. Естественно называть *важным* курьера, передающего заявки от сотрудника к кому-либо, кто подчиняется его приказам (или в обратном направлении).

Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения			Необх. подзадачи
		n	k	Дополнительно	
1	29	$2 \leqslant n \leqslant 100$	$2 \leqslant k \leqslant 100$	—	
2	12	$2 \leqslant n \leqslant 4000$	$2 \leqslant k \leqslant 1000$	—	1
3	7	$2 \leqslant n \leqslant 10^5$	$2 \leqslant k \leqslant 1000$	—	1-2
4	8	$2 \leqslant n \leqslant 10^5$	$2 \leqslant k \leqslant 5000$	—	1-3
5	10	$2 \leqslant n \leqslant 10^5$	$2 \leqslant k \leqslant 50\,000$	Число переходов между любым сотрудником и директором не превышает 20	
6	12	$2 \leqslant n \leqslant 10^5$	$2 \leqslant k \leqslant 50\,000$	Каждый курьер является важным	
7	12	$2 \leqslant n \leqslant 10^5$	$2 \leqslant k \leqslant 50\,000$	—	1-6
8	10	$2 \leqslant n \leqslant 2 \cdot 10^5$	$2 \leqslant k \leqslant 2 \cdot 10^5$	—	1-7

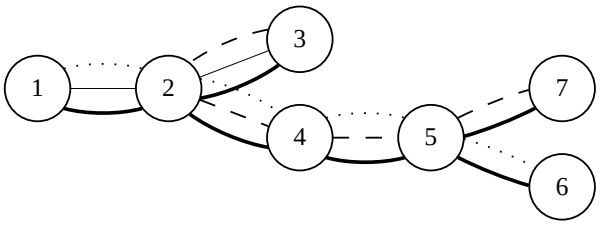
Примеры

twopaths.in	twopaths.out
4 2 1 2 2 1 3 1 4	1 2 1
4 2 1 2 3 1 2 3 4	0 1 2
7 3 1 2 2 4 5 5 1 3 3 7 6 1	2 2 3
4 3 1 2 3 1 4 4 1 1 4	3 2 1

Пояснения к примерам

В первом примере имеется два курьера. Первый курьер доставляет завки от сотрудника 1 сотруднику 3 или обратно. Так, доставляя завку от сотрудника 1 к сотруднику 3, он сначала передает заявку от сотрудника 1 сотруднику 2, а затем от сотрудника 2 сотруднику 3. Второй курьер доставляет завки от сотрудника 1 сотруднику 4 или обратно. Например, доставляя заявку от сотрудника 4 сотруднику 1, он сначала передаёт её от сотрудника 4 сотруднику 2, а затем от сотрудника 2 сотруднику 1. Степень дублирования этой пары курьеров равна 1, так как для них общим является переход между сотрудником 1 (директором) и сотрудником 2.

Во втором примере также имеется два курьера, но их пути не пересекаются.



Третий пример изображён на рисунке выше. Жирными линиями обозначены переходы между сотрудниками. Тонкими линиями обозначен путь первого курьера между сотрудниками 1 и 3. Длинным пунктиром обозначен путь второго курьера между сотрудниками 3 и 7. Коротким пунктиром обозначен путь третьего курьера между сотрудниками 1 и 6.

В пути первого и второго курьера одновременно входит только переход между сотрудниками 2 и 3. В пути первого и третьего курьера одновременно входит только переход между сотрудниками 1 и 2. В пути второго и третьего курьера одновременно входят два перехода: между сотрудниками 2 и 4 и между сотрудниками 4 и 5. Таким образом, пара из курьеров с номерами 2 и 3 имеет наибольшую степень дублирования, равную 2.

В четвертом примере все курьеры переносят заявки между директором и сотрудником номер 4. Таким образом, все пары курьеров имеют степень дублирования равную 3. Разрешается вывести любую пару.