

Инверсии в графе

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

Возможно, вы знакомы с классической задачей: дан массив a из n чисел, необходимо найти количество пар индексов $1 \leq i < j \leq n$ таких, что $a_i > a_j$. В этой задаче предлагается рассмотреть аналогичную задачу для обходов графа.

Пусть дан **связный** граф. Определим два типа обходов, которые будут использоваться в задаче. Перед началом любого обхода массивы `pre` и `post` изначально пусты.

Прямым обходом назовём массив `pre`, полученный следующим алгоритмом:

1. Добавить номер текущей вершины в массив `pre` и отметить её как посещённую.
2. Пока существуют непосещённые соседи текущей вершины, выбирать среди них вершину с **минимальным номером** и рекурсивно запустить этот же алгоритм из неё.
3. Если непосещённых соседей больше нет, работа алгоритма для данной вершины завершается.

Обратным обходом назовём массив `post`, полученный следующим алгоритмом:

1. Отметить текущую вершину как посещённую.
2. Пока существуют непосещённые соседи текущей вершины, выбирать среди них вершину с **минимальным номером** и рекурсивно запустить этот же алгоритм из неё.
3. После того как все соседи текущей вершины были посещены (или их нет), добавить номер текущей вершины в массив `post`.

Так как граф связный, каждая вершина окажется в обоих обходах **ровно один раз**. Обозначим за `prePos[v]` позицию вершины v в массиве `pre`, а за `postPos[v]` — её позицию в массиве `post`.

Вам дан граф из n вершин и m рёбер. Необходимо определить количество пар вершин $1 \leq v < u \leq n$, таких, что их относительный порядок в прямом и обратном обходах совпадает. Это эквивалентно выполнению одного из двух условий:

1. `prePos[v] < prePos[u]` и `postPos[v] < postPos[u]`;
2. `prePos[v] > prePos[u]` и `postPos[v] > postPos[u]`.

Формат входных данных

В первой строке содержатся два целых числа n и m — количество вершин и количество рёбер соответственно ($1 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 2 \cdot 10^5$).

Далее следуют m строк, i -я из которых содержит два целых числа u_i, v_i — описание ребра графа ($1 \leq u_i, v_i \leq n$).

Гарантируется, что граф связный. Обратите внимание, что граф **может содержать петли и кратные рёбра**.

Формат выходных данных

В единственной строке выведите одно целое число — количество пар вершин, удовлетворяющих условию.

Система оценки

В задаче используется оценка по группам. Баллы за группу начисляются только при прохождении всех тестов группы. Группа тестируется только если все необходимые предыдущие группы были пройдены.

Обратите внимание, что в группах 1, 2, 3 и 4 граф является деревом. Под «путём с крайней вершиной 1» подразумевается дерево, в котором степень вершины 1 равна 1, а степени всех остальных вершин не превосходят 2 (то есть граф представляет собой простой путь, начинающийся в вершине 1).

Группа	Ограничения	Баллы	Доп. ограничения	Зависимые группы
0	Тесты из условия	0	—	—
1	$n, m \leq 1000$	10	$m = n - 1$ граф — путь с крайней вершиной 1	—
2	$n, m \leq 2 \cdot 10^5$	12	$m = n - 1$ граф — путь с крайней вершиной 1	1
3	$n, m \leq 1000$	15	$m = n - 1$	1
4	$n, m \leq 2 \cdot 10^5$	28	$m = n - 1$	1, 2, 3
5	$n, m \leq 1000$	13	—	0, 1, 3
6	$n, m \leq 2 \cdot 10^5$	22	—	0, 1, 2, 3, 4, 5

Примеры

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