

Содержание

Must have	2
Задача 10А. Любители Кошек [0.1, 256]	2
Задача 10В. Компоненты связности [0.1, 256]	3
Задачи здорового человека	4
Задача 10С. Глубинный путь [0.5, 256]	4
Задача 10D. Дерево [0.1, 256]	5
Задача 10Е. Связанность графа [0.1, 256]	6
Задача 10F. Поиск цикла [0.5, 256]	7
Для искателей острых ощущений	8
Задача 10G. Поиск пути на гриде [2, 256]	8
Задача 10H. Деление на плоскости [2, 256]	9

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же `stdin`), вывести ответ нужно в **стандартный поток вывода** (он же `stdout`).

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`.

Подни можно пользоваться **дополнительной библиотекой** (`optimization.h`).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет `vector-set-map-весь-STL`): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 10А. Любители Кошек [0.1, 256]

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке входного файла заданы числа n и m ($1 \leq n \leq 1000$, $1 \leq m \leq 30\,000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел a_i b_i , обозначающие, что a_i знаком с b_i . Информация об одном знакомстве может быть записана несколько раз, причем даже в разном порядке (как (x, y) , так и (y, x)).

Формат выходных данных

В выходной файл необходимо вывести количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

stdin	stdout
3 3 1 2 2 3 3 1	1

Подсказка по решению

Вы умеете решать за $\mathcal{O}(nm)$.

Умеете даже быстрее (перебирать не все варианты третьей вершины, а...)

Задача 10В. Компоненты связности [0.1, 256]

Вам задан неориентированный граф с N вершинами и M ребрами ($1 \leq N \leq 20\,000$, $1 \leq M \leq 200\,000$). В графе отсутствуют петли и кратные ребра.

Определите компоненты связности заданного графа.

Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа N и M . Каждая из следующих M строк содержит описание ребра — два целых числа из диапазона от 1 до N — номера концов ребра.

Формат выходных данных

На первой строке выходного файла выведите число L — количество компонент связности заданного графа. На следующей строке выведите N чисел из диапазона от 1 до L — номера компонент связности, которым принадлежат соответствующие вершины. Компоненты связности следует занумеровать от 1 до L произвольным образом.

Пример

stdin	stdout
4 2	2
1 2	1 1 2 2
3 4	

Подсказка по решению

Просто поиск в глубину. Не забывайте, что локально на больших тестах ваша программа может падать, если вы не увеличили стек. В системе всё ок.

Задачи здорового человека

Задача 10С. Глубинный путь [0.5, 256]

Дан ориентированный (по рёбрам можно ходить только в одну сторону) граф из n вершин и m рёбер. Найдите любой путь из вершины s в вершину t .

Формат входных данных

На первой строке числа n, m, s, t ($2 \leq n \leq 50\,000, 0 \leq m \leq 100\,000, 1 \leq s, t, n, s \neq t$). Следующие m строк содержат пары целых чисел a_i, b_i , описывающие ребро из a_i в b_i . Граф не содержит петель, но может содержать кратные рёбра.

Формат выходных данных

Если пути нет, выведите -1 . Иначе выведите вершины пути в порядке от s до t .

Если путей из s в t несколько, выведите любой.

Путь должен быть простым (вершины пути не повторяются).

Примеры

stdin	stdout
4 5 1 3 1 2 2 4 2 1 4 3 4 1	1 2 4 3
4 5 3 1 1 2 2 4 2 1 4 3 4 1	-1

Подсказка по решению

Путь мы умеем восстанавливать на обратном ходу рекурсии.

Задача 10D. Дерево [0.1, 256]

Дан неориентированный граф. Проверьте, является ли он деревом.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 100$). В следующих m строках заданы рёбра; i -я из этих строк содержит два целых числа u_i и v_i через пробел — номера концов i -го ребра ($1 \leq u_i, v_i \leq n$). Граф не содержит петель и кратных рёбер.

Формат выходных данных

В первой строке выходного файла выведите “YES”, если граф является деревом, и “NO” в противном случае.

Примеры

stdin	stdout
3 2 1 2 1 3	YES
3 3 1 2 2 3 3 1	NO

Подсказка по решению

Просто поиск в глубину.

Задача 10Е. Связанность графа [0.1, 256]

Дан граф, содержащий N вершин и M рёбер ($1 \leq N \leq 1000, 1 \leq M \leq 7000$). Требуется найти наименьшее число рёбер и эти рёбра, которые нужно добавить, чтобы граф стал связным.

Формат входных данных

Во входном файле записаны сначала числа N и M , затем идёт описание рёбер графа — M пар чисел, где каждая пара описывает начало и конец ребра.

Формат выходных данных

В первую строку вывести единственное число K — минимальное количество рёбер, которое нужно добавить. В следующих K строках выведите по 2 числа — начало и конец нового ребра.

stdin	stdout
3 1	1
2 1	1 3

Подсказка по решению

Простая задача.

Задача 10F. Поиск цикла [0.5, 256]

Дан ориентированный невзвешенный граф без петель и кратных рёбер. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

Формат входных данных

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N \leq 100\,000$, $M \leq 100\,000$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходных данных

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

Примеры

stdin	stdout
2 2 1 2 2 1	YES 1 2
3 3 1 2 2 3 1 3	NO

Подсказка по решению

Просто dfs.

Для искателей острых ощущений

Задача 10G. Поиск пути на гриде [2, 256]

Дано прямоугольное поле $W \times H$. Некоторые клетки проходимы, через некоторые ходить нельзя. Из клетки можно ходить в соседние по ребру (слева, справа, сверху, снизу).

Нужно из клетки (x_1, y_1) найти любой (не обязательно кратчайший, даже не обязательно простой) путь в клетку (x_2, y_2) .

Формат входных данных

На первой строке W, H, x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 \leq W \leq 1000, 1 \leq y_1, y_2 \leq H \leq 1000$). Далее H строк, в каждой из которых по W символов. Символ “.” означает, что клетка проходима, а символ “*” означает, что по ней ходить нельзя.

Клетки (x_1, y_1) и (x_2, y_2) не совпадают и обе проходимы.

Формат выходных данных

Если пути не существует, выведите NO.

Иначе выведите YES и последовательность клеток (x_i, y_i) , в которой первая совпадает с клеткой (x_1, y_1) , а последняя с клеткой (x_2, y_2) .

Пример

stdin	stdout
4 2 1 1 4 2	YES 1 1 2 1 3 1 4 1 3 1 3 2 4 2
4 2 1 1 4 2 ..*. .*..	NO
4 2 1 1 4 2 ..*. *...	YES 1 1 2 1 2 2 3 2 4 2

Подсказка по решению

Тот же dfs и немного техники.

Задача 10Н. Деление на плоскости [2, 256]

Даны не более 1000 точек на плоскости. Координаты точек — целые числа от 0 до 10^9 . Нужно разделить точки на два множества таким образом, что максимальный из диаметров полученных множеств был как можно меньше.

Диаметром множества точек называется максимум попарных расстояний. Диаметр пустого множества равен нулю.

Формат входных данных

На первой строке количество точек n ($2 \leq n \leq 1000$). Следующие n точек содержат координаты точек. Гарантируется, что все точки различны.

Формат выходных данных

Выведите первое из двух множеств: на первой строке размер множества, на второй строке индексы точек, попавших в множество (точки нумеруются от 1 в том порядке, в котором даны). Если оптимальных ответов несколько, выведите любой.

Примеры

stdin	stdout
4 1 0 0 0 0 1 1 1	2 2 1
4 0 0 8 8 9 9 10 10	3 2 3 4

Подсказка по решению

Разобрана на практике.