Содержание

У вас не получается читать/выводить данные?
Воспользуйтесь примерами (c++) (python).

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 19A. Сумма расстояний [0.1, 256]

Дан связный неориентированный граф. Требуется найти сумму расстояний между всеми парами вершин.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leqslant n \leqslant 1000$, $0 \leqslant m \leqslant 10\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leqslant b_i, e_i \leqslant n$).

Гарантируется, что граф связен.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — сумму попарных расстояний между вершинами.

Пример

stdin	stdout
5 5 1 2 2 3 3 4 5 3 1 5	16

Замечание

Просто bfs. Не перемудрите.

Задача 19B. Расстояние между вершинами [0.2, 256]

Дан взвешенный неориентированный граф.

Требуется найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит два числа n и m — количество вершин и ребер графа соответственно. Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \leqslant s, t \leqslant n$, $s \neq t$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \leqslant b_i, e_i \leqslant n$, $0 \leqslant w_i \leqslant 100$).

$1 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t.

Если путь из s в t не существует, выведите `-1`.

Пример

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

Замечание

Дейкстра обыкновенная.

C++. `set<pair<int,int>>` зайдёт. `priority_queue` быстрее.

Эстеты вместо пар могут создать `struct Vertex {int id;}` и перегрузить `operator<`.

Обязательные задачи

Задача 19C. Расстояние между вершинами [0.1, 256]

Дан неориентированный взвешенный граф без петель и кратных рёбер. Найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит натуральные числа N, M, вторая строка содержит натуральные числа S и F ($N \leqslant 5\,000$, $M \leqslant 100\,000$, $1 \leqslant S, F \leqslant N$, $S \neq F$) — количество вершин и ребер графа а также номера вершин, длину пути между которыми требуется найти.

Следующие M строк по три натуральных числа b_i, e_i и w_i — номера концов i-ого ребра и его вес соответственно ($1 \leqslant b_i, e_i \leqslant n$, $0 \leqslant w_i \leqslant 100\,000$).

Формат выходных данных

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами S и F. Во второй строке через пробел выведите вершины на кратчайшем пути из S в F в порядке обхода.

Если путь из S в F не существует, выведите -1.

Пример

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

Замечание

Существует жизнь без кучи.

Матрица смежности в этой задачи крайне не эффективна из-за проблем с кешом.

Задача 19D. Стоимость проезда [0.1, 256]

Страна состоит из n городов и m дорог. Города пронумерованы числами от 1 до n. Город с номером s является столицей. Все дороги односторонние, проход по каждой дороге стоит ровно 1 золотой. Требуется найти минимальные стоимости проезда от каждого города до столицы.

Формат входных данных

В первой строке файла записаны три целых числа — n, s и m (количество городов, номер столичного города и количество дорог).

В следующих m строках записаны пары чисел. Пара чисел (a, b) означает, что есть дорога из города a в город b.

Ограничения: $1 \leqslant n \leqslant 10^5, 0 \leqslant m \leqslant 10^5$.

Формат выходных данных

Выведите n чисел — минимальные стоимости проезда от городов до столицы. Если от какого-то города не существует ни одного пути до столицы, выведите -1.

Пример

stdin	stdout
3 2 2 1 2 2 3	1 0 -1

Замечание

Казалось бы один bfs. Или нет? Или всё-таки да?

Задача 19E. Кратчайший путь двух коней [0.1, 256]

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке.

Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений.

Пример

stdin	stdout
a1 c2 c2 a1	1 b3 1 d4 2 a1 1 c2

Подсказка по решению

bfs?

Для искателей острых ощущений

Задача 19F. Мини-сокобан [0.1, 256]

Про оригинальную игру можно почитать здесь.

Вам дан лабиринт в виде поля $n \times n$, состоящий из проходимых клеток «.» и непроходимых клеток «#». Также в лабиринте есть три особые клетки: «*» — ящик, «s» — вы, «+» — место, куда нужно притащить ящик. За один ход вы можете сместиться на соседнюю по стороне клекту, если та проходима, или. Если на соседней клетке стоит ящик, а за ним в том же направлении есть одна пустая, вы можете толкнуть его на одну клетку и сместиться за ним. Сделайте так, чтобы ящик оказался ровно там, где нужно за минимальное число ходов.

Формат входных данных

На первой строке число n ($2 \leqslant n \leqslant 10$) — размер лабиринта. Следующие n строк содержат по n символов «.», «#», «*», «s», «+». Символы «*», «s», «+» в лабиринте встречаются ровно по одному разу.

Формат выходных данных

На первой строке выведите минимальное число ходов, чтобы поставить ящик на нужную позицию. При этом, где окажитесь вы, неважно. Если же ящик доставить невозможно, выведите -1. Если возможно, в следующей строке выведите кратчайшую последовательность ходов, каждый ход кодируйте буквой «l» (left, влево), «r» (right, вправо), «u» (up, вверх), «d» (down, влево). Если при очередном ходе происходит движение ящика, используйте заглавную букву. Если оптимальных ответов несколько, выведите любой.

Примеры

stdin	stdout
3 s#. +*. ...	6 ddrruL
2 s. +*	-1
10 s..#.....+ ...######. #......... #.##...... #...###### #.#......# #.#..##..# #...###*.# #....#...# .....#####	78 drdddrrdrrrrdrddlUUruLLLLd ldlluuurrDDDlddrruLLdlUUUU UUUruulldRurDldRRRRRRRdrUU

Замечание

В последнем примере ответ – одна строка, но для удобства чтения разделена на три части.

Подсказка по решению

Это всё ещё поиск в ширину, но теперь по сложному состоянию. Вам важно, где находится ящик, и где находитесь вы сами. Для восстановление ответа в ссылках назад удобно запоминать и предыдущее состояние, и куда шли, и толкали ли ящик.

Задача 19G. Грязь [0.1, 256]

— Здравствуйте! Могу я поговорить с Петровым? Алё, милый, привет... ты знаешь, у нас дома небольшая авария произошла... Но твой компьютер не пострадал, не волнуйся. Но теперь там немного грязно. Ну, то есть очень грязно. Но ты не волнуйся, я приготовила тебе твои болотные сапоги, у входа стоят. А грязь я уберу, как будет свободное время. Когда? Ну, наверное, когда в отпуск пойду. А, ну когда вернёмся из Турции. А, ну значит в следующий отпуск, но обязательно уберу. А пока я у мамы поживу. И ты, кстати, тоже можешь. Ну, как хочешь, я же не заставляю... Только пока я не убрала, ты там грязь не разводи, сильно сапогами по грязи не шлёпай и когда по чистому ходишь, сапоги снимай и тапочки обувай, я их тоже возле входа поставила, ты их бери с собой, когда идёшь по грязи и переобувай. А когда по чистому идёшь, бери сапоги, там грязь в разных местах. Программисты, как известно, не самые трудолюбивые люди, поэтому убирать грязь не станут. Но переобувать болотные сапоги каждый раз, когда переходишь от грязного пола к чистому и наоборот — удовольствие ниже среднего, уж лучше пройти лишние несколько метров. Чтобы прожить время до следующего отпуска с комфортом, надо срочно выработать способ добираться из одной точки квартиры с минимальным количеством переобуваний по пути, ну а уж среди них, конечно, выбрать самый короткий.

Формат входных данных

Мультитест. Каждый тест задаётся следующим образом.

В первой строке даны два целых числа M и N — размеры квартиры (в у.е.). $1 \leqslant N, M \leqslant 500$. Два целых числа во второй строке — координаты компьютера (в у.е.), а два целых числа в третьей строке — координаты холодильника (тоже в у.е.). Далее идут M строк по N символов в каждой — план квартиры. На плане 1 означает чистое место, 2 — грязное, 0 — стена или непроходимая грязь. Переходить можно только на клетки, имеющие общую вершину с данной, при переходе с чистой на грязную и наоборот надо переобуваться. Холодильник и компьютер находятся не в клетках, помеченных нулём. Левая верхняя клетка плана имеет координаты $(1, 1)$.

Формат выходных данных

Для каждого теста на отдельной строке выведите:

Длину кратчайшего пути (количество преодоленных квадратиков, включая начальный и конечный) с минимальным количеством переобуваний, и, через пробел, количество переобуваний (переобувание проходит при переходе с грязного на чистое и наоборот). Если пройти к холодильнику невозможно, вывести числа 0 0.

Пример

stdin	stdout
3 7 1 1 3 7 1200121 1212020 1112021 1 1 1 1 1 1 1	8 4 1 0

Подсказка по решению

Есть решение bfs-ом. Но любите запихивать, могут попихать Дейкстру.

Задача 19H. Лифтостроитель Эдуард [0.1, 256]

Эдуард работает инженером в компании «Нетривиальные лифты». Его очередное задание — разработать новый лифт для небоскрёба из h этажей.

У Эдуарда есть идея-фикс: он считает, что четырёх кнопок должно хватать каждому. Его последнее конструктивное предложение предполагает следующие кнопки:

- Подняться на a этажей вверх
- Подняться на b этажей вверх
- Подняться на c этажей вверх
- Вернуться на первый этаж

Исходно лифт находится на первом этаже. Пассажир использует три первые кнопки, чтобы попасть на нужный этаж. Если пассажир пытается переместиться на этаж, которого не существует, то есть нажать одну из первых трёх кнопок на этаже со слишком большим номером, лифт не перемещается.

Чтобы доказать, что план достоин реализации, Эдуард хочет подсчитать количество этажей, до которых возможно доехать с его помощью.

Формат входных данных

В первой строке записано целое число h — количество этажей небоскрёба ($1 \leqslant h \leqslant 10^{18}$).

Во второй строке записаны целые числа a, b и c — параметры лифта ($1 \leqslant a, b, c \leqslant 100\,000$).

Формат выходных данных

Выведите одно целое число — количество этажей, доступных с первого с помощью лифта.

Пример

stdin	stdout
15 4 7 9	9