Содержание

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

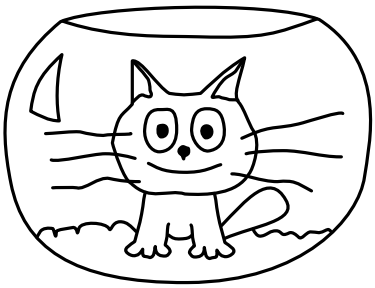
Обратите внимание на `GNU C++` компиляторы с суффиксом `inc`.
Подни можно пользоваться дополнительной библиотекой (`optimization.h`).
То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.
И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): пример.

Для тех, кто хочет разобраться, как всё это работает.
Короткая версия быстрого ввода-вывода (тык) и короткая версия аллокатора (тык).

Must have

Задача 9A. Продавец аквариумов [0.1, 256]

Продавец аквариумов для кошек хочет объехать n городов, посетив каждый из них ровно один раз. Помогите ему найти кратчайший путь.



Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leqslant n \leqslant 13$) — количество городов. Следующие n строк содержат по n чисел — длины путей между городами.

В i-й строке j-е число — $a_{i,j}$ — это расстояние между городами i и j ($0 \leqslant a_{i,j} \leqslant 10^6$; $a_{i,j} = a_{j,i}$; $a_{i,i} = 0$).

Формат выходных данных

В первой строке выходного файла выведите длину кратчайшего пути. Во второй строке выведите n чисел — порядок, в котором нужно посетить города.

Начинать и заканчивать путь можно в любом городе.

Пример

stdin	stdout
5 0 183 163 173 181 183 0 165 172 171 163 165 0 189 302 173 172 189 0 167 181 171 302 167 0	666 1 3 2 5 4

Подсказка по решению

Разобрано на лекции.

Задача 9B. Быстрое пересечение множеств [1.5, 256]

Даны N множеств. Множества занумерованы целыми числами от 1 до N. Для каждого множества $i = 1..N$ нужно найти такое множество $j = 1..N, j \neq i$, что их непохожесть минимальна. Непохожестью двух множеств A и B называется количество элементов, присутствующих ровно в одном из множеств A и B.

Формат входных данных

На первой строке целое число N от 2 до 10^4 — количество множеств. Далее собственно множества. Каждое множество задается следующим образом: сперва целое число k от 0 до 32 — размер множества, далее k целых чисел от 0 до 31 — элементы множества. Все элементы множества различны.

Формат выходных данных

Выведите N строк, в i-й строке выведите номер j — номер множества, которое вы считаете наименее непохожим на i-е), и собственно "непохожесть" данных множеств. Если для некоторого i существует несколько оптимальных j, выведите любое.

Пример

stdin	stdout
6 6 1 2 3 4 5 6 4 1 2 3 4 4 1 2 3 4 6 0 1 2 3 4 5 4 31 30 29 28 3 1 30 31	2 2 3 0 2 0 1 2 6 3 5 3

Подсказка по решению

Здесь нужна некая битовая магия и решение за N^2.

Задачи здорового человека

Задача 9C. Отметки на подмножествах 2 [0.3, 256]

Рассмотрим множество $\mathcal{S}$, состоящее из n элементов — натуральных чисел 1, 2, ..., n.

Сперва отметим несколько подмножеств $\mathcal{S}$, а также все подмножества этих подмножеств.

Затем снимем все отметки, если они есть, с нескольких подмножеств $\mathcal{S}$, а также со всех их подмножеств.

Найдите количество отмеченных подмножеств после всех этих операций.

Формат входных данных

В первой строке входного файла заданы через пробел три целых числа n, x и y. Следующие x строк содержат описания подмножеств, отмеченных на первом шаге, по одному на строке; также были отмечены все подмножества этих подмножеств. Наконец, последние y строк входного файла содержат описания подмножеств, с которых сняли отметки на втором шаге, по одному на строке; также были сняты отметки со всех их подмножеств. Описание каждого подмножества имеет вид $k\ a_1\ a_2\ \ldots\ a_k$, где k — количество элементов данного подмножества ($0 \leqslant k \leqslant n$), а a_i — сами элементы (a_i попарно различны, $1 \leqslant a_i \leqslant n$). Элементы могут быть перечислены в любом порядке.

Формат выходных данных

В первой строке выходного файла выведите одно число — количество отмеченных подмножеств после всех описанных операций.

Ограничения

- $1 \leqslant n \leqslant 20$
- $0 \leqslant x,\ y \leqslant 10\,000$

Примеры

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

Пояснения к примерам

В первом примере на первом шаге ставится отметка на подмножество {1} и на пустое подмножество, на втором шаге с пустого подмножества снимается отметка.

Во втором примере отметок нет.

В третьем примере на первом шаге отмеченными оказываются следующие шесть подмножеств: {}, {1}, {2}, {3}, {1, 2} и {2, 3}. На втором шаге снимаются отметки с трёх подмножеств {}, {1} и {3}.

Задача 9D. Гномы и число топсортов [0.3, 256]

Каждый вечер семья гномов ужинает за длинным столом. Все гномы сидят за одной стороной стола в определенном порядке. Разные места ценятся гномами по-разному, поскольку один конец стола находится ближе к кухне, и всем известно, что сидящий ближе к кухне получит еду раньше. Более того, некоторые гномы считают себя более важными, чем некоторые другие, и поэтому должны сидеть ближе к кухне.

Каждый вечер порядок, в котором гномы сидят за столом, должен быть новым. Посчитайте, сколько дней гномы смогут ужинать вместе.

Формат входных данных

Первая строка входного файла содержит два числа n и m ($1 \leqslant n \leqslant 19$), n — количество гномов, m — количество зависимостей. Каждая из следующих m строк содержит два целых числа x_i, y_i ($1 \leqslant x_i, y_i \leqslant n$); это означает, что гном x_i считает себя более важным, чем гном y_i. Каждая такая зависимость дается не более одного раза. Гномы пронумерованы целыми числами от 1 до n.

Формат выходных данных

Выведите единственное число — количество возможных рассадок гномов за столом.

Примеры

stdin	stdout
3 1 1 2	3

Подсказка по решению

Разобрано на практике.

Мы умеем решать за $2^n \cdot n^2$ и $2^n \cdot n$.

Задача 9E. Крышки [1, 256]

У программиста Петрова есть хобби — собирать крышки от пивных бутылок. Ничего странного, он знает еще с сотню программистов, которые очень уважают пиво. Да, они тоже собирают крышки. Не все из них, конечно, но некоторые. Если честно, то часть своих крышек он просто купил, уже без бутылок. Да, это не совсем спортивно, зато коллекция теперь более солидная. Одна вот беда — не хватает ему для полноты коллекции еще нескольких редких крышек. Он даже нашел в Интернете программистов, которые согласны продать их ему. Некоторые даже продают крышки сразу наборами, с большой скидкой. Почему продают, да еще со скидкой? А вы попробуйте объяснить жене, какая польза от пивных крышек. Она же не программист. Осталось выбрать оптимальные предложения. Если объяснить жене зачем надо хранить крышки еще возможно, то почему на них надо тратить деньги — точно не поймет. Поэтому надо купить как можно дешевле. Петров выписал на бумажку все варианты и задумался. Купить сразу все не получится, никаких денег не хватит. Тогда надо купить самые необходимые, но подешевле. Да уж, без программы тут не обойдешься...

Формат входных данных

В первой строке записано число N — количество недостающих Петрову крышек ($1 \leqslant N \leqslant 20$). Далее идет N строк — цена, за которую можно купить эту крышку, если покупать ее отдельно. В следующей строке стоит число M ($0 \leqslant M \leqslant 100$) — количество предложений по продаже наборов, содержащих нужные ему крышки. Далее идет M строк описывающих эти наборы. В каждой строке первое число — цена набора, второе — количество крышек в наборе, далее перечислены номера крышек (каждый номер от 1 до N), которые в этот набор входят. Номера в наборе не повторяются. Все цены — положительные числа, не превосходящие 2000. В последней строке перечислен минимальный набор крышек, который Петров намерен купить в любом случае.

Формат выходных данных

Выведите минимальную сумму, необходимую Петрову, чтобы купить все крышки из приведённого в последней строке набора.

Примеры

stdin	stdout
4 10 11 12 13 3 17 2 1 3 25 3 2 3 4 15 2 3 4 3 1 3 4	25

Подсказка по решению

Разобрано на практике.

Мы умеем решать за $2^n \cdot m$.

Для искателей острых ощущений

Задача 9F. Удаление скобок — 2 [0.2, 256]

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

Формат входных данных

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

Формат выходных данных

Выведите строку максимальной длины, являющуюся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов. Если возможных ответов несколько, выведите любой из них.

Примеры

stdin	stdout
([)]	[]
{([(]{)})]	([]{})
]{)[	

Подсказка по решению

Отрезки?

Задача 9G. Самый длинный путь 22 [1 sec, 256 mb]

В данном ориентированном графе найдите самый длинный путь такой, что каждая вершина графа встречается в нём не более одного раза.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и m ($1 \leqslant n \leqslant 22$, $0 \leqslant m \leqslant 1000$). В следующих m строках заданы рёбра графа в формате u_i v_i — номера начальной и конечной вершин ребра i, соответственно. Граф может содержать петли и кратные рёбра.

Формат выходных данных

В первой строке выходного файла выведите длину искомого пути l. Во второй строке выведите $l + 1$ число через пробел — вершины пути в порядке обхода. Если оптимальных ответов несколько, можно вывести любой из них.

Примеры

stdin	stdout
3 3 1 2 2 3 3 1	2 1 2 3
4 6 1 2 2 1 2 3 2 4 3 2 4 2	2 1 2 4
5 3 3 2 2 2 1 5	1 3 2

Подсказка по решению

Здесь не будут заходить решения за $2^n n^2$, нужно использовать битовую магию, чтобы получить $2^n n$. Можно придумать, можно прочесть конспект ПМИ.

Задача 9H. Справедливый дележ [1.5 sec, 256 mb]

— Я хотел честно, — сказал Балаганов, собирая деньги с кровати, — по справедливости.

В коробке от сигарет «Кавказ», отнятой у Корейко, было всего $1 \le N \le 15$ купюр, каждая номиналом $1 \leqslant a_i \leqslant 10^3$. Разделить надо было на $1 \le K \le 100$ частей, причём известно, что общая сумма делится на K. Метод деления «по справедливости» следующий: если разделить поровну не получается, то делить следует так, чтобы среднеквадратичное отклонение было минимальным.

Среднеквадратичное отклонение для данного способа дележа определяется следующим образом. Пусть i-й человек получил сумму денег, равную S_i. Обозначим за M среднее арифметическое сумм денег, полученных каждым: $M = (\sum_{i=1}^{K} S_i)/K$. Тогда среднеквадратичное отклонение можно вычислить так: $\sigma = \sqrt{(\sum_{i=1}^{K}(S_i - M)^2)/K}$.

— И как? — поинтересовался Остап.

— Сложно... — вздохнул Шура.

Попробуйте и вы разделить деньги «по справедливости».

Формат входных данных

В первой строке входного файла задано N — количество купюр и K — количество участников дележа. Далее, в следующей строке, заданы N целых чисел a_i — номиналы купюр.

Формат выходных данных

В первой строке выведите искомое среднеквадратичное отклонение с точностью до шести знаков после десятичной точки, в следующей строке выведите N чисел от 1 до K — номер участника дележа, которому досталась i-я купюра. Если оптимальных решений несколько, разрешается выводить любое.

Пример

stdin	stdout
4 3 1 2 3 6	1.414214 2 2 1 3

Замечание

Существует решение за $\mathcal{O}(3^n)$, но за $\mathcal{O}(3^n n)$ тоже зайдёт.