

## Содержание

<b>Разделяй, властвуй и FFT</b>	<b>2</b>
Задача A1. FFT и манная кашка [1 sec, 256 mb]	2
Задача A2. Деление многочленов [1 sec, 256 mb]	3
<b>Суффиксный автомат</b>	<b>4</b>
Задача B1. Суффиксный автомат [1 sec, 256 mb]	4
Задача B3. Помогите, спасите! [1 sec, 256 mb]	5
Задача B4. LZSS encoding [2 sec, 256 mb]	6
Задача B5. Подстроки-4 [1 sec, 256 mb]	7
<b>Линейное программирование</b>	<b>8</b>
Задача C1. Простая задача [0.2 sec, 256 mb]	8
Задача C2. Простая задача [0.2 sec, 256 mb]	9
Задача C3. Простая задача с хорошими тестами [0.2 sec, 256 mb]	10
Задача C4. Road times [0.5 sec, 256 mb]	11
<b>Паросочетания</b>	<b>12</b>
Задача D1. Рабочее расписание [0.2 sec, 256 mb]	12

---

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на компилятор GNU C++14 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

## Разделяй, властвуй и FFT

### Задача A1. FFT и манная кашка [1 сек, 256 mb]

Даны два многочлена

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$$

$$B(x) = b_m x^m + b_{m-1} x^{m-1} + \dots + b_0$$

Найти  $C(x) = A(x)B(x)$

#### Формат входных данных

$n, a_n, a_{n-1}, \dots, a_0$

$m, b_m, b_{m-1}, \dots, b_0$

$0 \leq n, m < 2^{16}, |a_i|, |b_j| \leq 9$

$a_n \neq 0, b_m \neq 0$

#### Формат выходных данных

Выведите коэффициенты  $C$  в том же формате.

#### Примеры

mulpoly.in				mulpoly.out				
2	1	0	2	3	2	3	4	6
1	2	3						

### Задача A2. Деление многочленов [1 сек, 256 mb]

Даны два многочлена с коэффициентами из  $\mathbb{Z}/7\mathbb{Z}$ . Старший коэффициент обоих не равен нулю. Нужно поделить их с остатком.

#### Формат входных данных

Каждая из двух строк задаёт описание многочлена. Многочлен  $a_k x^k + \dots + a_2 x^2 + a_1 x + a_0$  описывается числом  $k$  ( $0 \leq k \leq 50\,000$ ) и  $k + 1$  числами от 0 до 6:  $a_k, \dots, a_2, a_1, a_0$ .

#### Формат выходных данных

На первой строке многочлен-частное. На второй строке многочлен-остаток. Выводите многочлены в том же формате. Если многочлен – тождественный ноль, для него  $k = 0$ .

#### Примеры

divpoly.in	divpoly.out
3 1 1 1 1 1 1 1	2 1 0 1 0 0
3 1 1 3 1 2 1 1 1	1 1 0 1 2 1
8 2 1 2 1 2 1 2 1 2 4 1 2 3 4 5	4 2 4 2 5 2 3 3 1 3 6

## Суффиксный автомат

Задача В1. Суффиксный автомат [1 сек, 256 mb]

Или зачёт, или автомат.

---

Ганнибал Ректор

Дана строка. Постройте её суффиксный автомат.

### Формат входных данных

Строка длины от 1 до 100 000, состоящая из маленьких латинских букв.

### Формат выходных данных

На первой строке число состояний автомата и число рёбер. Следующие строки содержат рёбра в формате “откуда” “куда” “символ на ребре”. Далее число терминальных состояний и строка, содержащая все терминальные состояния в произвольном порядке. Начальным состоянием автомата должно быть состояние номер один.

### Примеры

automaton.in	automaton.out
ababb	7 9 1 2 a 1 7 b 2 3 b 3 4 a 3 6 b 4 5 b 5 6 b 7 4 a 7 6 b 3 6 7 1

**Задача В3. Помогите, спасите! [1 sec, 256 mb]**

Дана строка. Найдите для каждого её префикса количество различных подстрок в нём.

**Формат входных данных**

В единственной строке входных данных содержится непустая строка  $S$ , состоящая из  $N$  ( $1 \leq N \leq 2 \cdot 10^5$ ) маленьких букв английского алфавита.

**Формат выходных данных**

Выведите  $N$  строк, в  $i$ -й строке должно содержаться количество различных подстрок в  $i$ -м префиксе строки  $S$ .

**Примеры**

keepcounted.in	keepcounted.out
aabab	1 2 5 8 11
atari	1 3 5 9 14

### Задача В4. LZSS encoding [2 sec, 256 mb]

Алиса хочет отправить сообщение Бобу. Она хочет зашифровать сообщение, используя оригинальный метод шифрования. Сообщение – строка  $S$ , состоящая из  $N$  строчных английских букв.

$S[a..b]$  означает подстроку  $S$  от  $S[a]$  до  $S[b]$  ( $0 \leq a \leq b < N$ ). Если первые  $i$  букв уже зашифрованы, Алиса найдёт такие  $(j, k)$ :  $s[j..j+k] = s[i..i+k]$ ,  $k \geq 0$ ,  $0 \leq j < i$ ,  $k = \max$ . Если несколько  $j$  дают максимальное  $k$ , Алиса выберет минимальное  $j$ . Если  $k > 0$  Алиса добавит пару  $\langle j, k \rangle$  в шифр и увеличит  $i$  на  $k$ , иначе Алиса добавит  $-1$  и ASCII код буквы  $S[i]$  в шифр и увеличит  $i$  на  $1$ .

Очевидно шифр начнёт с  $-1$ , далее будет ASCII код символа  $S[0]$ . Помогите Алисе реализовать её метод шифрования.

#### Формат входных данных

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 50$ ). Следующие  $T$  строк содержат сообщения для шифровки, каждое длины от  $1$  до  $10^5$ , состоящие из строчных английских букв. Гарантируется, что суммарная длина всех сообщений не превосходит  $2 \cdot 10^6$ .

#### Формат выходных данных

Для каждого теста на отдельной строке выведите “Case #X:”, где  $X$  – номер теста, нумерация с  $1$ . Далее выведите шифр, в каждой строке по два целых числа через пробел.

#### Примеры

lzss.in	lzss.out
2	Case #1:
aaaaaa	-1 97
aaaaabbbbbaaabbc	5 0
	Case #2:
	-1 97
	4 0
	-1 98
	4 5
	5 2
	-1 99

**Задача В5. Подстроки-4 [1 сек, 256 mb]**

Даны  $K$  строк из маленьких латинских букв. Найдите их наибольшую общую подстроку.

**Формат входных данных**

В первой строке число  $K$  ( $1 \leq K \leq 10$ ). Далее  $K$  строк длины от 1 до 200 000.

**Формат выходных данных**

Наибольшая общая подстрока.

**Примеры**

substr4.in	substr4.out
3 abacaba myscabarchive acabistrue	cab

## Линейное программирование

### Задача C1. Простая задача [0.2 sec, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ x_1 \geq 0 \\ \dots \\ x_n \geq 0 \\ c_1x_1 + \dots + c_nx_n \rightarrow \max \end{cases}$$

#### Формат входных данных

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  — количество переменных и количество уравнений ( $1 \leq n, m \leq 5$ ). Следующие  $m$  строк содержат по  $n + 1$  целому числу:  $a_{i1}, \dots, a_{in}, b_i$ . Следующая строка содержит  $n$  целых чисел:  $c_1, \dots, c_n$ . Все числа во входном файле не превышают 100 по модулю.

#### Формат выходных данных

Выведите одно число: максимальное значение  $c_1x_1 + \dots + c_nx_n$ . Гарантируется, что решение существует и максимальное значение достигается.

#### Пример

simple.in	simple.out
2 2 1 2 3 2 1 3 1 1	2.0
1 1 1 2 -2	0.0
1 1 4 5 3	3.75

**Задача C2. Простая задача [0.2 sec, 256 mb]**

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ x_1 \geq 0 \\ \dots \\ x_n \geq 0 \\ c_1x_1 + \dots + c_nx_n \rightarrow \max \end{cases}$$

**Формат входных данных**

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  — количество переменных и количество уравнений ( $1 \leq n, m \leq 5$ ). Следующие  $m$  строк содержат по  $n + 1$  целому числу:  $a_{i1}, \dots, a_{in}, b_i$ . Следующая строка содержит  $n$  целых чисел:  $c_1, \dots, c_n$ . Все числа во входном файле не превышают 100 по модулю.

**Формат выходных данных**

Выведите одно число: максимальное значение  $c_1x_1 + \dots + c_nx_n$ . Если решения нет, выведите “No solution”. Если можно получить сколь угодно большое значение, выведите “Unbounded”.

**Пример**

simple2.in	simple2.out
2 2 1 2 3 2 1 3 1 1	2.0
2 1 -1 -1 0 1 1	Unbounded
2 1 1 1 -1 1 1	No solution

**Задача С3. Простая задача с хорошими тестами [0.2 сек, 256 mb]**

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ x_1 \geq 0 \\ \dots \\ x_n \geq 0 \\ c_1x_1 + \dots + c_nx_n \rightarrow \max \end{cases}$$

**Формат входных данных**

Входной файл содержит один или несколько тестов. На первой строке число тестов, далее сами тесты. Каждый тест описывается следующим образом. Первая строка теста содержит два целых числа:  $n$  и  $m$  — количество переменных и количество неравенств ( $1 \leq n, m \leq 5$ ). Следующие  $m$  строк содержат по  $n + 1$  целому числу:  $a_{i1}, \dots, a_{in}, b_i$ . Следующая строка содержит  $n$  целых чисел:  $c_1, \dots, c_n$ . Все числа во входном файле целые и не превышают 100 по модулю.

**Формат выходных данных**

Для каждого теста выведите одну строку.

Выведите максимальное значение  $c_1x_1 + \dots + c_nx_n$ , пробел, двоеточие, пробел, сами числа  $x_1x_2 \dots x_n$ . Все числа выводите с максимальной точностью. Если решения нет, выведите “No solution”. Если можно получить сколь угодно большое значение, выведите “Unbounded”.

**Пример**

simple3.in	simple3.out
4	2.0 : 1.0 1.0
2 2	Unbounded
1 2 3	No solution
2 1 3	1.5 : 0 1.5
1 1	
2 1	
-1 -1 0	
1 1	
2 1	
1 1 -1	
1 1	
2 2	
1 2 3	
2 1 3	
-10 1	

### Задача C4. Road times [0.5 sec, 256 mb]

Дорожная сеть страны – ориентированный граф. У каждого ребра есть длина, целое число от 1 до 1000, и ограничение на максимальную скорость, вещественное число от 30 до 60, в километрах в час. Длины вам известны, а ограничения – нет. Известно, что когда водитель такси берётся доставить пассажира из вершины  $a$  в вершину  $b$ , он осуществляет перевозку строго по кратчайшему пути между вершинами и едет с максимальной допустимой скоростью. Длина пути – сумма длин рёбер. Также известно, что  $\forall a, b \exists$  *единственный* кратчайший путь из  $a$  в  $b$ .

Ваша задача – по длинам рёбер и уже сделанным поездкам  $a_i b_i time_i$  оценить минимальное и максимальное время в пути между вершинами  $c_j d_j$

#### Формат входных данных

На первой строке число вершин  $n$  ( $1 \leq n \leq 30$ ).

Вершины нумеруются числами от 0 до  $n - 1$ .

Следующие  $n$  строк содержат матрицу  $n \times n$  длин дорог. Дороги односторонние.

Отсутствие дорог обозначено числом  $-1$ , длины дорог целые от 1 до 1000.

Всего не более 100 дорог.

Далее идёт число  $r$  ( $1 \leq r \leq 100$ ) и  $r$  уже известных маршрутов  $a_i b_i time_i$ ,  $time_i \in \mathbb{R}$ .

Затем число запросов  $q$  ( $1 \leq q \leq 100$ ) и  $q$  строк  $c_j d_j$ .

#### Формат выходных данных

Для каждого запроса выведите четыре числа – откуда, куда, min время, max время.

#### Примеры

roadtimes.in	roadtimes.out
3	0 1 50.0 80.0
0 50 -1	1 2 40.0 70.0
55 0 40	1 0 55.0 110.0
-1 40 0	
1	
0 2 120	
3	
0 1	
1 2	
1 0	

## Паросочетания

### Задача D1. Рабочее расписание [0.2 сек, 256 mb]

Дан неориентированный граф из  $N$  вершин и нескольких рёбер.  
Найти максимальное паросочетание.

*Здесь можно прочесть оригинальную легенду.*

#### Формат входных данных

На первой строке число вершин  $N$  ( $1 \leq N \leq 222$ ).

Далее строки, содержащие рёбра.

#### Формат выходных данных

На первой строке число вершин, покрытых максимальным паросочетанием.

На следующих строках рёбра паросочетания.

Если есть несколько максимальных паросочетаний, выведите любое.

#### Примеры

schedule.in	schedule.out
3	2
1 2	1 2
2 3	
1 3	

#### Подсказка по решению

Алгоритм Эдмондса сжатия соцветий. Реализация Габова.