

Содержание

03.Base [3/3]	3
Задача 03A. Экзамен по истории [1 sec, 256 mb]	3
Задача 03B. Поиск [1 sec, 256 mb]	4
Задача 03C. Быстрый поиск в массиве [1 sec, 256 mb]	5
03.Advanced [4/6]	6
Задача 03D. Вербочки [0.5 sec, 256 mb]	6
Задача 03E. Сумма трёх [6 sec, 256 mb]	7
Задача 03F. Минимальный максимум [2.5 sec, 256 mb]	8
Задача 03G. Сумма не без разнообразия [1 sec, 256 mb]	9
Задача 03H. Для любителей статистики [1 sec, 256 mb]	10
Задача 03I. Коровы – в стойла! [0.3 sec, 256 mb]	11
03.Hard [0/4]	12
Задача 03J. La cucaracha [0.1 sec, 256 mb]	12
Задача 03K. K-Best [0.4 sec, 256 mb]	13
Задача 03L. Лифт [0.5 sec, 256 mb]	14
Задача 03M. Многочлен [0.5 sec, 256 mb]	15

Общая информация:

Вход в констест: <http://contest.yandex.ru/contest/2836/>

Дедлайн на задачи: 9 дней, до 2016-10-01 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <https://compscicenter.ru/courses/algorithms-1/2016-autumn/>

Семинары ведут Сергей Копелиович (burunduk30@gmail.com, vk.com/burunduk1) и Алексей Кладов (aleksey.kladov@gmail.com).

В каждой условии указан таймлимит для C/C++.

Таймлимит для Java примерно в 2-3 раза больше.

Таймлимит для Python примерно в 6 раз больше.

C++:

Быстрый ввод-вывод.

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html Более подробно про ввод-вывод.

http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Java:

Быстрый ввод-вывод.

http://acm.math.spbu.ru/~sk1/algo/input-output/java/java_common.html

03.Base [3/3]

Задача 03А. Экзамен по истории [1 sec, 256 mb]

Даны два списка дат.

Найти количество дат во втором списке, которые присутствуют в первом.

Формат входных данных

На первой строке длина первого списка N ($1 \leq N \leq 15\,000$).

Следующие N строк содержат целые числа от 1 до 10^9 – даты из первого списка.

Далее длина второго списка M ($1 \leq M \leq 10^6$).

Следующие M строк содержат целые числа от 1 до 10^9 – даты из второго списка.

Формат выходных данных

Одно число – ответ на задачу.

Примеры

history.in	history.out
2	2
1054	
1492	
4	
1492	
65536	
1492	
100	

Замечание

Задачу можно сдавать как угодно..

Задача 03В. Поиск [1 sec, 256 mb]

В этой задаче нужно уметь выяснять, содержится ли число в последовательности.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и k ($1 \leq n \leq 300\,000$, $1 \leq k \leq 300\,000$). Во второй строке задана последовательность из n отсортированных целых чисел a_1, a_2, \dots, a_n , записанных через пробел ($1 \leq a_i \leq 10^9$). В третьей строке записаны запросы — k целых чисел b_1, b_2, \dots, b_k записанных через пробел, в порядке возрастания ($1 \leq b_j \leq 10^9$).

Формат выходных данных

В выходной файл выведите k строк. В j -ой строке выведите “YES”, если число b_j содержится в последовательности $\{a_i\}$, и “NO” в противном случае.

Примеры

find2.in	find2.out
3 3	NO
2 3 5	YES
1 2 3	YES
3 4	YES
1 2 2	YES
1 2 4 5	NO
	NO

Замечание

Потренируйтесь писать свой бинпоиск.

Потренируйтесь пользоваться стандартным в своём языке, сдайте задачу ещё раз.

Задача 03С. Быстрый поиск в массиве [1 сек, 256 mb]

Дан массив из N целых чисел. Все числа от -10^9 до 10^9 .

Нужно уметь отвечать на запросы вида “Сколько чисел имеют значения от L до R ?”.

Формат входных данных

Число N ($1 \leq N \leq 10^5$). Далее N целых чисел.

Затем число запросов K ($1 \leq K \leq 10^5$).

Далее K пар чисел L, R ($-10^9 \leq L \leq R \leq 10^9$) — собственно запросы.

Формат выходных данных

Выведите K чисел — ответы на запросы.

Пример

find3.in	find3.out
5	5 2 2 0
10 1 10 3 4	
4	
1 10	
2 9	
3 4	
2 2	

Замечание

Потренируйтесь писать свой бинпоиск. Напишите один бинпоиск, а не два.

Потренируйтесь пользоваться стандартным в своём языке, сдайте задачу ещё раз.

03.Advanced [4/6]

Задача 03D. Веребочки [0.5 sec, 256 mb]

С утра шел дождь, и ничего не предвещало беды. Но к обеду выглянуло солнце, и в лагерь заглянула СЭС. Пройдя по всем домикам и корпусам, СЭС вынесла следующий вердикт: бельевые веревки в жилых домиках не удовлетворяют нормам СЭС. Как выяснилось, в каждом домике должно быть ровно по одной бельевой веревке, и все веревки должны иметь одинаковую длину. В лагере имеется N бельевых веревок и K домиков. Чтобы лагерь не закрыли, требуется так нарезать данные веревки, чтобы среди получившихся веревочек было K одинаковой длины. Размер штрафа обратно пропорционален длине бельевых веревок, которые будут развешены в домиках. Поэтому начальство лагеря стремится максимизировать длину этих веревочек.

Формат входных данных

В первой строке заданы два числа — N ($1 \leq N \leq 10\,001$) и K ($1 \leq K \leq 10\,001$). Далее в каждой из последующих N строк записано по одному числу — длине очередной бельевой веревки. Длина веревки задана в сантиметрах. Все длины лежат в интервале от 1 сантиметра до 100 километров включительно.

Формат выходных данных

В выходной файл следует вывести одно целое число — максимальную длину веревочек, удовлетворяющую условию, в сантиметрах. В случае, если лагерь закроют, выведите 0.

Пример

ropes.in	ropes.out
4 11	200
802	
743	
457	
539	

Замечание

Разобрана на семинаре

Задача 03Е. Сумма трёх [6 сек, 256 mb]

Даны три массива целых чисел A, B, C и целое число S .

Найдите такие i, j, k , что $A_i + B_j + C_k = S$.

Формат входных данных

На первой строке число S ($1 \leq S \leq 10^9$). Следующие три строки содержат описание массивов A, B, C в одинаковом формате: первое число задает длину n соответствующего массива ($1 \leq n \leq 15\,000$), затем заданы n целых чисел от 1 до 10^9 — сам массив.

Формат выходных данных

Если таких i, j, k не существует, выведите единственное число -1 . Иначе выведите на одной строке три числа i, j, k . Элементы массивов нумеруются с нуля. Если ответов несколько, выведите лексикографически минимальный.

Примеры

threesum.in	threesum.out
3 2 1 2 2 3 1 2 3 1	0 1 1
10 1 5 1 4 1 3	-1
5 4 1 2 3 4 3 5 2 1 4 5 3 2 2	0 1 2

Замечание

Разобрана на семинаре

Задача 03F. Минимальный максимум [2.5 сек, 256 mb]

Даны n нестрого возрастающих массивов A_i и m нестрого убывающих массивов B_j . Все массивы имеют одну и ту же длину l . Далее даны q запросов вида (i, j) , ответ на запрос – такое k , что $\max(A_{ik}, B_{jk})$ минимален. Если таких k несколько, можно вернуть любое.

Формат входных данных

На первой строке числа n, m, l ($1 \leq n, m \leq 900; 1 \leq l \leq 3000$). Следующие n строк содержат описания массивов A_i . Каждый массив описывается перечислением l элементов. Элементы массива – целые числа от 0 до $10^5 - 1$. Далее число m и описание массивов B_j в таком же формате. Массивы и элементы внутри массива нумеруются с 1. На следующей строке число запросов q ($1 \leq q \leq n \cdot m$). Следующие q строк содержат пары чисел i, j ($1 \leq i \leq n, 1 \leq j \leq m$).

Формат выходных данных

Выведите q чисел от 1 до l – ответы на запросы.

Примеры

minmax.in	minmax.out
4 3 5	3
1 2 3 4 5	4
1 1 1 1 1	3
0 99999 99999 99999 99999	5
0 0 0 0 99999	4
5 4 3 2 1	3
99999 99999 99999 0 0	1
99999 99999 0 0 0	1
12	1
1 1	4
1 2	4
1 3	3
2 1	
2 2	
2 3	
3 1	
3 2	
3 3	
4 1	
4 2	
4 3	

Замечание

≈ 37 мегабайт ввода. Аккуратно.

Задача 03G. Сумма не без разнообразия [1 сек, 256 mb]

Задана последовательность целых чисел A_1, A_2, \dots, A_N .

Необходимо выбрать из нее подпоследовательность из подряд стоящих чисел A_i, A_{i+1}, \dots, A_j так, чтобы она содержала не менее K различных чисел, и сумма $S = A_i + A_{i+1} + \dots + A_j$ была максимальной.

Формат входных данных

Первая строка ввода содержит целые числа N и K ($1 \leq K \leq N \leq 200\,000$).

Вторая строка содержит N целых чисел A_1, A_2, \dots, A_N ($|A_i| \leq 1\,000\,000\,000$).

Формат выходных данных

В первой строке необходимо вывести максимальное возможное значение суммы S . Во второй строке выведите индексы первого и последнего элементов найденной оптимальной подпоследовательности. Если существует несколько решений, подойдет любое из них.

Если не существует подпоследовательностей, удовлетворяющих решению задачи, выведите одну строку со словом “IMPOSSIBLE” (без кавычек).

Примеры

threemax.in	threemax.out
7 3 -99 1 2 -100 3 2 3	-89 2 7
3 2 1 1 1	IMPOSSIBLE

Замечание

Разобрана на семинаре.

Хеш-таблица получила TL? Напишите свою или сделайте сжатие координат.

Задача 03Н. Для любителей статистики [1 сек, 256 mb]

Вы никогда не задумывались над тем, сколько человек за год перевозят трамваи города с десятиллионным населением, в котором каждый третий житель пользуется трамваем по два раза в день?

Предположим, что на планете Земля n городов, в которых есть трамваи. Любители статистики подсчитали для каждого из этих городов, сколько человек перевезено трамваями этого города за последний год. Из этих данных была составлена таблица, в которой города были отсортированы по алфавиту. Позже выяснилось, что для статистики названия городов несущественны, и тогда их просто заменили числами от 1 до n . Поисковая система, работающая с этими данными, должна уметь быстро отвечать на вопрос, есть ли среди городов с номерами от l до r такой, что за год трамваи этого города перевезли ровно x человек. Вам предстоит реализовать этот модуль системы.

Формат входных данных

В первой строке дано целое число n , $0 < n < 70\,000$. В следующей строке приведены статистические данные в виде списка целых чисел через пробел, i -е число в этом списке — количество человек, перевезенных за год трамваями i -го города. Все числа в списке положительны и не превосходят $10^9 - 1$. В третьей строке дано количество запросов q , $0 < q < 70\,000$. В следующих q строках перечислены запросы. Каждый запрос — это тройка целых чисел l , r и x , записанных через пробел ($1 \leq l \leq r \leq n$, $0 < x < 10^9$).

Формат выходных данных

Выведите строку длины q , в которой i -й символ равен 1, если ответ на i -й запрос утвердителен, и 0 в противном случае.

Пример

queries.in	queries.out
5	10101
123 666 314 666 434	
5	
1 5 314	
1 5 578	
2 4 666	
4 4 713	
1 1 123	

Замечание

Разобрана на семинаре

Задача 031. Коровы – в стойла! [0.3 sec, 256 mb]

На прямой расположены стойла, в которые необходимо расставить коров так, чтобы минимальное расстояние между коровами было как можно больше.

Формат входных данных

В первой строке вводятся числа N ($2 < N < 10001$) — количество стойл и K ($1 < K < N$) — количество коров. Во второй строке задаются N натуральных чисел в порядке возрастания — координаты стойл (координаты не превосходят 10^9).

Формат выходных данных

Выведите одно число — наибольшее возможное допустимое расстояние.

Примеры

cows.in	cows.out
5 3 1 2 3 100 1000	99

Замечание

Разобрана на семинаре

03.Hard [0/4]

Задача 03J. La cucaracha [0.1 sec, 256 mb]

Каждую полночь в квартире ученого Васи начинается ужас. Сотни . . . , о нет! ТЫСЯЧИ тараканов вылазят из каждой дырки к его обеденному столу, уничтожая все крошки и объедки! Вася ненавидит тараканов. Он очень долго думал и сделал Супер-ловушку, которая привлекает всех тараканов в большой зоне после активации. Он планирует активировать ловушку сегодня ночью. Но есть проблема. Эта очень эффективная ловушка с её очень большой зоной работы поглощает огромное количество энергии. Так что, Вася планирует минимизировать время работы этой ловушки. Он собрал информацию о всех местах, в которых живут тараканы. Также он заметил, что все тараканы двигаются только по линиям его скатерти с постоянной скоростью (мы можем предположить, что эта скорость равна 1, так что таракан расположенный в одной из секций, может за 1 единицу времени переместиться на любую соседнюю секцию (по вертикали или горизонтали)). Вася решил активировать его ловушку в одной из секций. Когда ловушка активирована, все тараканы будут двигаться к секции, содержащей ловушку, так быстро, как только смогут. Поэтому в любой момент времени после активации тараканы двигаются к секции, в которой находится ловушка, максимально уменьшая расстояние до неё. Если есть два пути с одинаковым расстоянием, то таракан выберет любой. Напишите программу для Васи, которая выбирает секцию, минимизирующую время, необходимое для уничтожения всех тараканов. Конечно, ваша программа будет считать, что скатерть будет плоскостью с декартовой системой координат и секции — точки с целыми координатами.

Формат входных данных

В первой строке входного файла содержится число мест, в которых живут тараканы N ($1 \leq N \leq 10000$). Следующие N строк содержат x и y — координаты мест, в которых живут тараканы (целые числа не больше 10^9 по абсолютному значению).

Формат выходных данных

Вам необходимо вывести только два целых числа x и y , не превосходящие по модулю 10^9 , — координаты секции, которая минимизирует время работы. Если есть более одно решение — выведите любое из них.

Пример

cucarach.in	cucarach.out
2	2 2
1 1	
3 3	

Задача 03К. K-Best [0.4 sec, 256 mb]

У Демьяны есть n драгоценностей. Каждая из драгоценностей имеет ценность v_i и вес w_i . С тех пор, как её мужа Джонни уволили в связи с последним финансовым кризисом, Демьяна решила продать несколько драгоценностей. Для себя она решила оставить лишь k лучших. Лучших в смысле максимизации достаточно специфического выражения: пусть она оставила для себя драгоценности номер i_1, i_2, \dots, i_k , тогда максимальной должна быть величина

$$\frac{\sum_{j=1}^k v_{i_j}}{\sum_{j=1}^k w_{i_j}}$$

Помогите Демьяне выбрать k драгоценностей требуемым образом.

Формат входных данных

На первой строке n и k ($1 \leq k \leq n \leq 100\,000$).

Следующие n строк содержат пары целых чисел v_i, w_i ($0 \leq v_i \leq 10^6, 1 \leq w_i \leq 10^6$, сумма всех v_i не превосходит 10^7 , сумма всех w_i также не превосходит 10^7).

Формат выходных данных

Выведите k различных чисел от 1 до n — номера драгоценностей. Драгоценности нумеруются в том порядке, в котором перечислены во входных данных. Если есть несколько оптимальных ответов, выведите любой.

Пример

kbest.in	kbest.out
3 2	1 2
1 1	
1 2	
1 3	

Замечание

Где-то здесь зарыт бинпоиск.

Задача 03L. Лифт [0.5 sec, 256 mb]

Высокое здание, состоящее из N этажей, оснащено только одним лифтом. Парковка находится ниже фундамента здания, что соответствует одному этажу ниже первого. Этажи пронумерованы от 1 до N снизу вверх. Про каждый этаж известно количество человек, желающих спуститься на лифте на парковку. Пусть для i -го этажа эта величина равна A_i . Известно, что лифт не может перевозить более C человек одновременно, а также то, что на преодоление расстояния в один этаж (не важно вверх или вниз) ему требуется P секунд. Какое наибольшее количество человек лифт может перевезти на парковку за T секунд, если изначально он находится на уровне парковки?

Формат входных данных

В первой строке входного файла содержатся целые числа N, C, P, T ($1 \leq N \leq 100$, $1 \leq C \leq 10^9$, $1 \leq P \leq 10^9$, $1 \leq T \leq 10^9$). Вторая строка содержит последовательность N целых чисел A_1, A_2, \dots, A_N ($0 \leq A_i \leq 10^9$). Сумма всех значений последовательности не превосходит 10^9 .

Формат выходных данных

Выведите наибольшее количество человек, которое лифт успеет перевезти на парковку.

Пример

lift.in	lift.out
4 5 2 15 0 1 2 3	3
4 5 2 18 0 1 2 3	5
3 2 1 9 1 1 1	3

Задача 03М. Многочлен [0.5 sec, 256 mb]

Каждый новый русский хочет дать своим детям всё самое лучшее. В частности, лучшее образование. Например, Колян попросил учителя математики научить его сына решать не только квадратные уравнения, но также кубические, четвёртой и вообще любой степени. Учитель знает, что уравнения степени выше пятой не могут быть решены в радикалах в общем виде. Но решение уравнений до пятой степени также достаточно сложная задача. Лучше проверить решения, используя компьютер. Вот где нужна ваша помощь.

Формат входных данных

Первая строка содержит степень многочлена N ($1 \leq N \leq 5$). В следующей $N + 1$ строке находятся целые числа ($-100 \leq a_i \leq 100$, $a_0 \neq 0$). Строка $i + 2$ содержит i -й коэффициент многочлена $a_0x^n + a_1x^{n-1} + \dots + a_n$.

Формат выходных данных

Выведите все вещественные корни многочлена, учитывая их кратность. Корни должны быть выведены в порядке возрастания. Точность должна быть не менее 10^{-6} .

Примеры

polynom.in	polynom.out
2	1
1	1
-2	
1	

Замечание

Погрешность. Аккуратно.