

Содержание

Закрепляем лекцию	2
Задача А. Две лесопилки [0.1 sec, 256 mb]	2
Задача В. Петя и прямоугольники [0.5 sec, 256 mb]	3
Получаем удовольствие от дорешивания	4
Задача С. Новая модель телефона [0.1 sec, 256 mb]	4
Задача D. Мультипликативная сумма [0.5 sec, 256 mb]	5
Задача Е. Калила и Димна на лесозаготовках [0.2 sec, 256 mb]	6
Задача F. Сервера [0.6 sec, 256 mb]	7
Задача G. Order-Preserving Codes [1 sec, 256 mb]	8
Задача H. Командный пункт [3 sec, 256 mb]	9
Задача I. Копилка [4 sec, 256 mb]	10
Задача J. Выстрелы по стенам [0.5 sec, 256 mb]	11
Самая обычная динамика	13
Задача K. Мостостроение [0.3 sec, 256 mb]	13

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.

Перед вами подборка задач на такие оптимизации динамики, как

- convex hull trick
- divide & conquer
- оптимизация Кнута
- два указателя

Автор подборки – Сергей Копелиович (burunduk30@gmail.com).

Закрепляем лекцию

Задача А. Две лесопилки [0.1 sec, 256 mb]

Спонсор этой задачи — банк баянов 2005-го года. Банк баянов — мы помним всё!

От вершины до подножья холма растет N старых деревьев. Районная администрация решила в санитарных целях срубить эти деревья, а чтобы снизить стоимость мероприятия перевезти все древесину на лесопилки. Деревья могут быть перевезены только в одном направлении — вниз. У подножья холма находится лесопилка, а также две дополнительные лесопилки могут быть построены на холме вдоль дороги. Вам предстоит определить, где наиболее выгодно построить эти лесопилки, чтобы минимизировать стоимость транспортировки древесины. Перевозка 1 килограмма древесины на 1 метр стоит 1 копейку.

Формат входных данных

Первая строка входного файла содержит натуральное число N — количество деревьев ($1 \leq N \leq 20\,000$). Деревья занумерованы от 1 до N начиная с вершины холма. Следующие N линий содержат по два целых числа w_i и d_i ($1 \leq w_i, d_i \leq 10\,000$) — вес дерева номер i и расстояние между деревьями i и $i+1$. Последнее из этих чисел (d_n) задает расстояние от нижнего дерева до лесопилки.

Формат выходных данных

Выведите единственное число — минимальную стоимость сплава деревьев вниз по реке.

Пример

stdin	stdout
9	26
1 2	
2 1	
3 3	
1 1	
3 2	
1 6	
2 1	
1 2	
1 1	

Пояснение к примеру

В примере выгодно поставить лесопилки у деревьев с номерами 3 и 6.

Задача В. Петя и прямоугольники [0.5 sec, 256 mb]

Спонсор этой задачи — Харьковские сборы 2014. Хорошие были сборы...

Маленький Петя очень любит прямоугольники. Петя дал маме список прямоугольников, которые он хочет получить в подарок на Новый Год. Каждый прямоугольник характеризуется w и высотой h . Мама хочет сделать Пете приятное и купить все прямоугольники из его списка. Мама отправилась в магазин и узнала, что цена одного прямоугольника равна его площади. К ее счастью, в магазине действует предновогодняя акция, позволяющая покупать прямоугольники не по одному, а сразу наборами. Стоимость одного набора равна ширине самого широкого прямоугольника, умноженной на высоту самого высокого прямоугольника из этого набора. Обратите внимание, что поворачивать прямоугольники (тем самым меняя местами ширину и высоту) нельзя. Помогите маме Пети купить все прямоугольники из списка ее сына, потратив на это наименьшее количество денег.

Формат входных данных

В первой строке записано число N ($1 \leq N \leq 200\,000$) — количество прямоугольников в списке Пети. В каждой из следующих N строк записаны по 2 целых положительных числа, не превышающих 10^6 — ширина и высота очередного прямоугольника.

Формат выходных данных

Выведите одно число — наименьшее количество денег, которое может потратить мама чтобы купить Пете все прямоугольники из его списка.

Примеры

stdin	stdout
4 100 1 15 15 20 5 1 100	500
5 1 10 2 20 3 30 4 40 10 1	170

Получаем удовольствие от дорешивания

Задача С. Новая модель телефона [0.1 сек, 256 mb]

Спонсор этой задачи — тренировки СПб ГУ.

Компания Gnusmas разработала новую модель мобильного телефона. Основное достоинство этой модели — ударопрочность: её корпус сделан из особого сплава, и телефон должен выдерживать падение с большой высоты.

Компания Gnusmas арендовала n -этажное здание и наняла экспертов, чтобы те при помощи серии экспериментов выяснили, с какой высоты бросать телефон можно, а с какой — нельзя. Один эксперимент заключается в том, чтобы бросить телефон с какого-то этажа и посмотреть, сломается он от этого или нет. Известно, что любой телефон этой модели ломается, если его сбросить с x -го этажа или выше, где x — некоторое целое число от 1 до n , включительно, и не ломается, если сбросить его с более низкого этажа. Задача экспертов заключается в том, чтобы узнать число x и передать его рекламному отделу компании.

Задача осложняется тем, что экспертам предоставлено всего k образцов новой модели телефона. Каждый телефон можно бросать сколько угодно раз, пока он не сломается; после этого использовать его для экспериментов больше не удастся.

Подумав, эксперты решили действовать так, чтобы минимизировать максимально возможное количество экспериментов, которое может потребоваться произвести. Чему равно это количество?

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа n и k — количество этажей в здании и количество предоставленных телефонов ($1 \leq n \leq 100\,000$, $0 \leq k < n$).

Формат выходных данных

В выходной файл выведите единственное число — минимальное количество экспериментов, которое потребуется совершить, чтобы узнать число x и использовать не более k телефонов. Если решить задачу невозможно, выведите вместо этого -1 .

Примеры

stdin	stdout
4 2	2
4 1	3

Замечание

В первом примере сначала следует бросить телефон со второго этажа. Если он сломается, то второй бросок следует сделать с первого этажа. В случае поломки станет известно, что $x = 1$. Иначе мы узнаем, что $x = 2$.

Если же при броске со второго этажа телефон не сломался, бросим телефон с третьего этажа. При поломке будет ясно, что $x = 3$. Иначе из условия $1 \leq x \leq 4$ следует, что $x = 4$.

Всего будет сделано два эксперимента. В них будет использовано не более чем два телефона.

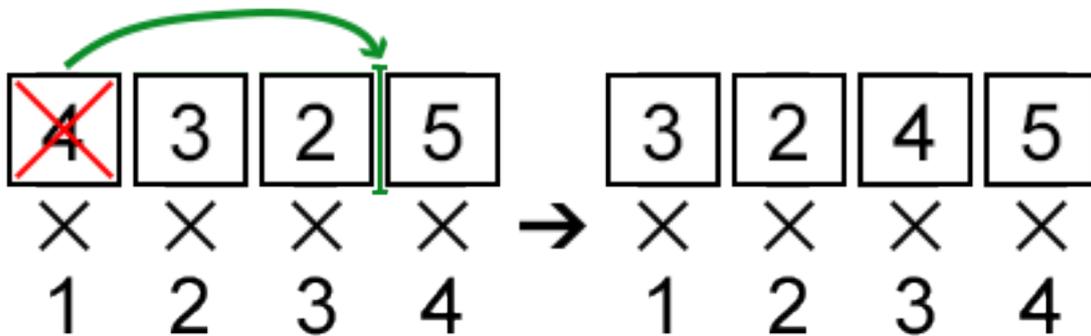
Во втором примере следует сначала бросить единственный данный нам телефон с первого этажа, если он не сломается, то со второго, а если опять не сломается, то с третьего. При первой же поломке мы узнаем точное значение x . Если после трёх бросков телефон так и не сломался, то $x = 4$.

Задача D. Мультипликативная сумма [0.5 sec, 256 mb]

Спонсоры этой задачи — Codeforces. Codeforces — мечты сбываются!

Хотя Блейк и является начальником Криса, это никак не мешает их дружбе. Обычно по вечерам они собираются вместе в баре и обсуждают там увлекательные задачи, в которых требуется максимизировать какую-нибудь величину. В этот раз им попалась особенная задача...

Пусть дан массив a длины n . Назовём *характеристикой* массива число $c = \sum_{i=1}^n a_i \cdot i$ — сумму произведений i -го числа в массиве a на число i . Разрешается **ровно один раз** сделать следующую операцию: выбрать некоторый элемент массива и переставить его в любое место в массиве. В частности, разрешается переставить элемент в начало, в конец или поставить его обратно на ту же позицию. Необходимо получить массив с максимально возможным значением характеристики.



Формат входных данных

В первой строке входных данных записано одно число n ($2 \leq n \leq 200\,000$) — размер массива a .

Во второй строке содержится n целых чисел a_i ($|a_i| \leq 1\,000\,000$) — элементы массива a .

Формат выходных данных

Выведите единственное целое число — максимально возможное значение характеристики массива a , если можно переставить ровно один элемент.

Примеры

stdin	stdout
4 4 3 2 5	39
5 1 1 2 7 1	49
3 1 1 2	9

Замечание

В первом примере можно удалить первый элемент из массива и вставить его перед третьим (перед числом 5), таким образом получив максимальный ответ: $3 \cdot 1 + 2 \cdot 2 + 4 \cdot 3 + 5 \cdot 4 = 39$.

Во втором примере максимальный ответ можно получить, удалив пятый элемент из массива и вставив его перед третьим: $1 \cdot 1 + 1 \cdot 2 + 1 \cdot 3 + 2 \cdot 4 + 7 \cdot 5 = 49$.

Задача E. Калила и Димна на лесозаготовках [0.2 сек, 256 mb]

Спонсор этой задачи — Codeforces round 189. Codeforces — мечты сбываются!

Калила и Димна — два шакала. Они живут в огромных джунглях. Однажды шакалы решили устроиться на завод лесозаготовки и подработать.

Управляющий завода хочет, чтобы они отправились в джунгли и срубили n деревьев высотой a_1, a_2, \dots, a_n . Для этого Калила и Димна купили цепную пилу в магазине. Каждый раз, когда они используют пилу на дереве номер i , они уменьшают высоту этого дерева на единицу. Каждый раз Калила и Димна должны заправить пилу для использования. Цена заправки зависит от того, какие деревья полностью спилены (дерево считается полностью спиленным, если его высота равна 0). Если максимальный идентификатор полностью срубленного дерева равняется i (первоначально это дерево имело высоту a_i), то цена заправки пилы равняется b_i . Если ни одно дерево не срублено полностью, то заправлять пилу запрещается. Изначально пила заправлена. Известно, что для каждого $i < j$, $a_i < a_j$ и $b_i > b_j$, а также $b_n = 0$ и $a_1 = 1$.

Калила и Димна хотят полностью срубить все деревья с минимальными затратами. Они ждут Вашей помощи! Поможете?

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 10^5$). Во второй строке записано n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$). В третьей строке записано n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$).

Гарантируется, что $a_1 = 1$, $b_n = 0$, $a_1 < a_2 < \dots < a_n$ и $b_1 > b_2 > \dots > b_n$.

Формат выходных данных

В единственной строке должна быть записана минимальная стоимость вырубания всех деревьев.

Примеры

stdin	stdout
5 1 2 3 4 5 5 4 3 2 0	25
6 1 2 3 10 20 30 6 5 4 3 2 0	138

Задача F. Сервера [0.6 sec, 256 mb]

Спонсор этой задачи — всесибирская олимпиада 2009. Новосибирск – город мечты!

Компьютерная сеть в некотором доме строилась по принципу присоединения нового компьютера к последнему из уже подключенных. Никакие два компьютера, будучи подключенными в сеть, между собой дополнительно никак не связывались. Таким образом, в сеть были объединены последовательно N компьютеров. Соседи обменивались информацией между собой, но в какой-то момент поняли, что им нужны прокси-серверы. Компьютерное сообщество дома решило установить прокси-серверы ровно на K компьютеров. Осталось только решить, какие именно компьютеры выбрать для этой цели. Главным критерием является ежемесячная стоимость обслуживания серверами всех компьютеров.

Для каждого компьютера установлен тариф его обслуживания, выраженный в рублях за метр провода. Стоимость обслуживания одного компьютера каким-то сервером равна тарифу компьютера, умноженному на суммарную длину провода от этого компьютера до сервера, которым он обслуживается.

Ваша задача написать программу, которая выберет такие K компьютеров, чтобы установить на них прокси-серверы, что общие затраты на обслуживание всех компьютеров были бы минимальными

Формат входных данных

В первой строке входного файла записано два целых числа N и K — количество компьютеров в сети и количество прокси-серверов, которые нужно установить ($1 \leq K \leq N \leq 2000$).

Все компьютеры в сети пронумерованы числами от 1 до N по порядку подключения.

Во второй строке записано одно целое число T_1 — тариф обслуживания первого компьютера.

В следующих $N - 1$ строках записано через пробел по два целых неотрицательных числа L_i, T_i — информация об остальных компьютерах в сети по порядку номеров. L_i — длина провода, соединяющего i — компьютер с соседним с меньшим номером, T_i — тариф обслуживания данного компьютера ($2 \leq i \leq N$). Все L_i и T_i от 0 до 10^6 .

Формат выходных данных

В первую строку выходного файла необходимо вывести одно целое число — минимальную стоимость обслуживания всех компьютеров всеми серверами. Во второй строке должны быть записаны через пробел K номеров компьютеров, на которые необходимо установить серверы. При существовании нескольких вариантов размещения разрешается вывести любой.

Пример

stdin	stdout
3 1 10	19 1
2 2 3 3	
3 2 10	4 1 3
2 2 3 3	

Задача G. Order-Preserving Codes [1 sec, 256 mb]

Спонсор этой задачи — Андрей Станкевич и Petrozavodsk Camp. English – connecting people!

Binary code is a mapping of characters of some alphabet to the set of finite length bit sequences. For example, standard ASCII code is a fixed length code, where each character is encoded using 8 bits.

Variable length codes are often used to compress texts taking into account the frequencies of occurrence of different characters. Characters that occur more often get shorter codes, while characters occurring less often — longer ones.

To ensure unique decoding of variable length codes so called *prefix codes* are usually used. In a prefix code no code sequence is a proper prefix of another sequence. Prefix code can be easily decoded scanning the encoded sequence from left to right, since no code is the prefix of another, one always knows where the code for the current character ends and the new character starts.

Among prefix codes, the optimal code is known, so called Huffman code. It provides the shortest possible length of the text among all prefix codes that separately encode each character with an integer number of bits.

However, as many other codes, Huffman code does not preserve character order. That is, Huffman codes for lexicographically ordered characters are not necessarily lexicographically ordered.

In this problem you are asked to develop a prefix code that would be optimal for the given text among all order-preserving prefix codes. Code is called order-preserving if for any two characters the code sequence for the character that goes earlier in the alphabet is lexicographically smaller.

Since text itself is not essential for finding the code, only the number of occurrences of each character is important, only this data is given.

Формат входных данных

The first line of the input file contains n — the number of characters in the alphabet ($2 \leq n \leq 2000$). The next line contains n integer numbers — the number of occurrences of the characters in the text for which the code must be developed (numbers are positive and do not exceed 10^9). Characters are described in the alphabetical order.

Формат выходных данных

Output n bit sequences, one on a line — the optimal order-preserving prefix code for the described text.

Пример

stdin	stdout
5	00
1 8 2 3 1	01
	10
	110
	111

Задача Н. Командный пункт [3 сек, 256 mb]

Спонсор этой задачи — Petrozavodsk Winter Camp 2010-2011, Saratov Contest.

Полковник Кругляковски только что закончил свою последнюю кампанию. Сейчас он установил свой командный пост в центре прекрасной долины и хочет защитить себя и свою армию от возможных атак.

Чтобы добиться защищённости, нужно построить несколько наблюдательных пунктов. Кругляковски уже отправил своих лучших скаутов, которые по возвращении доложили ему список из n наиболее подходящих мест для постройки наблюдательных пунктов. Все n найденных мест располагаются на одинаковом расстоянии от командного пункта (то есть на окружности с центром в точке расположения командного поста). К своему сожалению, Кругляковски быстро понял, что ресурсов хватит на постройку только k наблюдательных пунктов. Кругляковски также планирует построить забор, соединив соседние наблюдательные пункты.

Пожалуйста, помогите Колонэлю выбрать k позиций для наблюдательных постов таким образом, чтобы максимизировать площадь, окружённую забором. При этом командный пункт должен оказаться внутри забора, или на границе. Вы можете предположить, что земля плоская, пренебречь размером командного пункта и наблюдательных постов. Забор — это выпуклый многоугольник с вершинами в наблюдательных постах.

Формат входных данных

Первая строка содержит два целых числа n и k ($3 \leq k \leq n \leq 1000$). Следующие n чисел содержат полярные углы (в радианах) возможных наблюдательных постов. Углы даны на отдельных строках. Углы задаются вещественными числами с не более чем четырьмя знаками после запятой. Все углы находятся в диапазоне от 0 до 2π . Все углы различны. Командный пост находится в точке $(0, 0)$.

Формат выходных данных

Выведите k различных целых чисел от 1 до n — индексы точек, которые вы используете для постройки наблюдательных постов. Точки нужно перечислить в порядке увеличения полярного угла. Если есть несколько оптимальных решений, выведите любое. Если же защитить командный пункт с помощью забора невозможно, выведите вместо этого 0 на отдельной строке.

Примеры

stdin	stdout
4 4 1.57 0 3.14 4.71	2 1 3 4
4 3 1.57 0 3.14 4.71	2 1 4

Задача I. Копилка [4 sec, 256 mb]

Спонсор этой задачи — чемпионат Урала и Битва Гигантов 2013.

Дракон хранит свои сбережения в копилке. Каждый день Дракон или кладёт ровно один доллар в копилку, или берёт из копилки сколько-то денег. Дракону интересно, могут ли его сбережения стать значительно больше, если бы он вкладывал в копилку больше.

Говоря более точно, Дракон задаёт вам серию вопросов: “Если вкладывать в копилку дополнительных z долларов каждый день, каково максимальное количество денег в копилке в период с дня a по день b ?”

Формат входных данных

Первая строка содержит числа n и m ($1 \leq n, m \leq 50000$) – число дней и количество вопросов. Следующая строка содержит n целых чисел s_1, s_2, \dots, s_n – количество денег в копилке после i -го дня ($0 \leq s_i \leq s_{i-1} + 1, s_0 = 0$). Следующие m строк содержат тройки чисел a_i, b_i, z_i : $1 \leq a_i \leq b_i \leq n$ и $1 \leq z_i \leq 10^9$. Учтите, что Дракон не властен над временем, поэтому каждый запрос нужно обрабатывать независимо от остальных, используя исходную последовательность s .

Формат выходных данных

Выведите m строк, содержащие ответы на вопросы.

Примеры

stdin	stdout
9 3	10
1 2 1 2 3 4 1 2 0	13
1 9 1	4
5 7 4	
4 4 2	

Подсказка по решению

Можно, конечно, воспользоваться свойством $s_{i+1} \leq s_i + 1$, но, используя изученные нами трюки, можно не сложно решить задачу без этого ограничения в общем случае.

Задача J. Выстрелы по стенам [0.5 sec, 256 mb]

*Спонсор этой задачи — Petrozavodsk Summer Camp 2005 и команда MSU SE.
Отдельное спасибо команде по изготовлению гробов.*

Производится испытание нового пистолета, который может производить выстрелы с различными скоростями пуль. В некоторые моменты времени происходят выстрелы из начала координат с определенными горизонтальными скоростями, и в некоторые другие моменты строят стены на горизонтальной площадке — невырожденные отрезки, лежащие на прямых, не проходящих через начало координат. При этом стены могут пересекаться. Для обработки результатов эксперимента необходимо определить, сколько времени летела каждая пуля. Пуля летит с постоянной скоростью. Пуля останавливается сразу при касании стены.

Формат входных данных

В начале каждой строки написано одно из трех слов: `shot`, `wall` или `end`. Число строк не превышает 50 000. После слова `shot` следуют две координаты скорости пули. Скорость пули не может равняться нулю. После слова `wall` следуют четыре числа — координаты начала и конца стены. Слово `end` является признаком окончания набора входных данных. Все координаты являются целыми числами, по модулю не превосходящими 10 000. Все события записаны в хронологическом порядке, и интервалы времени между событиями больше, чем время, за которое строится стена, и чем время, за которое пуля пролетает расстояние до ближайшей стены или за границу испытательного полигона.

Формат выходных данных

Для каждого выстрела вывести на отдельной строке одно число — время, которое пролетит пуля, с точностью до 10^{-6} . Если пуля не попадет ни в какую стену, то вместо числа вывести слово `Infinite`.

Пример

stdin	stdout
shot 1 0	Infinite
wall 1 0 0 1	0.50000000000000000000
shot 1 1	Infinite
shot -1 3	0.50000000000000000000
wall 1 0 -1 2	0.33333333333333333333
shot -1 3	2.00000000000000000000
wall 1 1 -1 1	0.05000000000000000000
shot -1 3	0.00020000000000000000
wall 2 3 2 -3	2.00000000000000000000
wall 3 -2 -3 -2	0.00100000000000000000
shot 1 -1	Infinite
shot 40 -39	0.00099950024987506247
shot 9999 -10000	1.00000000000000000000
shot -1 -1	0.50000000000000000000
shot -3000 -2000	1.00000000000000000000
shot -3001 -2000	0.90909090909090909091
shot -3000 -2001	0.43478260869565217391
shot 1 0	0.83333333333333333333
shot 1 1	2.00000000000000000000
wall -1 2 10 -10	3333.3333333333333333
shot -1 1	
shot 0 1	
shot 1 1	
shot 1 0	
shot 1 -1	
wall 0 -10000 -10000 0	
shot -2 -1	
end	

Самая обычная динамика

Задача К. Мостостроение [0.3 сек, 256 mb]

Спонсоры этой задачи — чемпионат СПбГУ и питерская школьная олимпиада 2009.

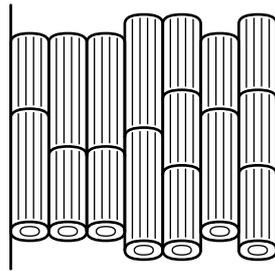
В деревне Зайкино регулярно идут проливные дожди, в результате чего речка Дубровка, которую обычно можно просто перешагнуть, выходит из берегов. Чтобы можно было перейти разлившуюся реку, планируется построить плавучий мост из брёвен, оставшихся от строительства бани бизнесмена, поселившегося неподалёку.

Все оставшиеся брёвна имеют одинаковую толщину. При этом есть x брёвен длины a и y брёвен длины b .

Построенный мост должен состоять из l рядов, каждый из которых составлен из одного или нескольких брёвен. Пилить брёвна нельзя, так как последняя пила утонула при разливе Дубровки.

Главный инженер хочет построить мост максимальной возможной ширины. Ширина моста определяется по минимальной ширине ряда брёвен в нём.

Например, если нужно построить мост из семи рядов, и при этом есть шесть брёвен длины 3 и десять брёвен длины 2, то можно построить мост ширины 5.



Формат входных данных

Ввод состоит из одного или нескольких тестовых случаев. Каждый тестовый случай состоит из пяти целых положительных чисел x , a , y , b и l . Каждое число не превосходит 500. Общее количество брёвен в каждом тестовом случае не меньше l .

Обозначим $d = \max(x, a, y, b, l)$.

Гарантируется, что сумма d по всем тестам не превосходит 5000.

Формат выходных данных

Для каждого тестового случая на отдельной строке выведите одно число — максимальную возможную ширину моста.

Пример

stdin	stdout
6 3 10 2 7	5
10 7 20 9 25	9
106 126 135 28 137	112