

Муниципальный этап
Всероссийской олимпиады школьников
по информатике
в 2018 – 2019 учебном году

Разборы решений и идеи тестов

**Муниципальный этап Всероссийской олимпиады
школьников по информатике
в 2018 – 2019 учебном году
9–11 класс**

Время выполнения задач — 4 часа

Ограничение по времени — 2 секунды на тест

Ограничение по памяти — 256 мегабайт

9–11.1. «Выпить чашечку кофе...». Большинство подданных императора находятся на станции-цитадели Колизеум. В том числе, и Яна и Савов — средне-статистические подданные Империи, отнюдь не самого высокого звания, которые зарабатывают себе на жизнь разгрузкой и загрузкой контейнеров с микросхемами «Ореон» и макросхемами «Стук». Но даже простые грузчики Империи могут позволить себе раз в день напитков с десертиком в одном из многочисленных кафе Колизеума. На каждом a -м этаже цитадели открыта кофейня «CofFix» (то есть на этажах $0, a, 2a, \dots$), а на каждом b -м — чайная «DifficulTea» (то есть на этажах $0, b, 2b, \dots$). Исторически сложилось так, что числа a и b взаимнопросты. Чтобы разнообразить свои трудовые будни, Яна и Савов решили как можно реже посещать один и тот же этаж. Помогите Яне и Савову посчитать, сколько дней они смогут ходить в питейные заведения на разные этажи, прежде чем им придется посетить какой-то этаж снова.

Формат входа: В первой строке через пробел заданы три целых числа n — номер последнего этажа в Колизеуме, a и b ($1 \leq a, b \leq n$, a и b взаимно просты).

Формат выхода: Выведите единственное целое число — максимальное число дней, в течение которые Яна и Савов при посещении питейных заведений могут ходить на различные этажи Колизеума.

Пример

Вход: Выход:

10 2 3 8

Система оценивания

<i>Подзадача</i>	<i>Баллы</i>	<i>Ограничения</i>	<i>Проверка</i>
1	20	$1 \leq n \leq 100$	полная
2	40	$1 \leq n \leq 10^5$	баллы
3	40	$1 \leq n \leq 10^9$	баллы

9–11.2. «Саботаж». Йи Гроег — обыкновенный, ничем не примечательный мальчик с планеты Рутнок, который, как и все юноши, мечтал о великих делах. По достижению совершеннолетия, Йи пошел в повстанческую Школу Подготовки Разведчиков, а после прохождения всех испытаний в ШПР Йи отправился работать на благо повстанцев и устроился на имперский склад на станции-цитадели Колизеум.

Не прошло и недели с момента прибытия, как Йи Гроег раздобыл ценные планы имперцев по поставке деталей техники в дни с номерами от 1 до n . Особенно ценными для повстанцев являются микросхемы «Ореон» и макросхемы «Стук». Гроег знает, что в i -й день на склад привозят A_i микросхем стоимостью p ЗЕТов и B_i макросхем стоимостью q ЗЕТов. (Микросхемы стоят дороже макросхем.) Также он узнал, что в конце дня все схемы отправляются на завод, который тщательно охраняется, и бесполезно даже пытаться украсть детали оттуда.

За один день Йи может вынести со склада не больше S схем — ровно столько вмещается в бардачке его космолета. Какой наибольший ущерб Гроег может нанести Империи своими действиями и что ему для этого необходимо сделать?

Формат входа: В первой строке входных данных находятся четыре числа — n , S , p , q — количество дней; количество деталей, которое вмещает бардачок; стоимость схем «Ореон» и «Стук», соответственно ($p > q$).

В следующих N строках перечислены пары чисел A_i и B_i — количество схем «Ореон» и «Стук», соответственно, которые привезут на склад утром i -го дня.

Формат выхода: В первой строке выведите единственное число — максимальная суммарная стоимость деталей (в ЗЕТах), которые Йи сможет похитить.

Далее выведите n строк, где в i -й строке через пробел перечислены два числа X_i , Y_i — количество микросхем «Ореон» и количество макросхем «Стук», которые он должен изъять со склада в i -й день.

Пример

Вход: Выход:

```
2 5 5 1 42
10 10 5 0
3 2 3 2
```

Система оценивания

Подзадача	Баллы	Ограничения	Проверка
1	10	$N = 3, 1 \leq p, q, S, A_i, B_i \leq 100$	полная
2	10	$1 \leq N, p, q, S, A_i, B_i \leq 100, A_i + B_i \leq S$	полная
3	10	$1 \leq N, p, q, S, A_i \leq 100, B_i = 0$	полная
4	40	$1 \leq N, p, q, S, A_i, B_i \leq 100$	баллы
5	30	$1 \leq N, S \leq 1000, 1 \leq p, q, A_i, B_i \leq 10^9$	баллы

9–11.3. «Праздник урожая Гречи». Сегодня на планете Ксвежи, население которой известно своими кулинарными умениями, отмечают Праздник урожая Гречи. Ашас, уроженец этой планеты, хоть и находится вдали от родины, не хочет оставаться в стороне. Поэтому он решил приготовить праздничное блюдо, самое вкусное, какое только можно сделать из имеющихся у него n ингредиентов. Каждый ингредиент имеет регистрационный номер в Великой Поваренной книге Ксвежи — последовательность десятичных цифр, а его вкусность v равна сумме цифр его номера. Вкусность же всего блюда определяется как вкусностью используемых ингредиентов, так и последовательностью их добавления: вкусность блюда — это сумма разностей вкусностей соседних ингредиентов, где из вкусности предыдущего ингредиента вычитается вкусность следующего. Вкусность блюда может быть и отрицательной! Помогите Ашасу найти последовательность добавления ингредиентов, дающую самое вкусное блюдо.

Формат входа: В первой строке содержится единственное целое число n — количество ингредиентов, которыми располагает Ашас. В следующих n строках содержится по единственной последовательности символов десятичных цифр — регистрационные номера ингредиентов. Суммарная длина всех последовательностей не превосходит 10^6 .

Формат выхода: В первой строке выведите вкусность самого вкусного блюда, которое можно получить. Во второй строке через пробел выведите последовательность регистрационных номеров ингредиентов в том порядке добавления в готовящееся блюдо, который даст самое вкусное блюдо. Если таких последовательностей несколько, выведите любую.

Пример 1

Вход: Выход:
 4 21
 456 90909 456 008910 1023
 008910
 1023
 90909

Система оценивания

<i>Подзадача</i>	<i>Баллы</i>	<i>Ограничения</i>	<i>Проверка</i>
1	20	$n = 2$, сумма длин не превосходит 100	полная
2	50	$n \leq 10^5$, названия состоят из одной цифры	баллы
3	30	$n \leq 10^5$, сумма длин слов не превосходит 10^6	баллы

9–11.4. «Побег с планеты». Сталкер Арю решил сбежать от повстанцев и стать частью Империи. Для этого ему необходимо добраться до космодрома планеты Рутнок. Он исследовал местность с помощью сканирующей прямой и обнаружил n защитных экранов, каждый из которых представляет собой отрезок. По данным сканирующей прямой Арю находится в точке с координатами $(0, 0)$, космодром — в точке $(X, 0)$, а каждый из защитных экранов перпендикулярен прямой между Арю и космодромом. При этом Арю знает, что никакие два экрана не расположены на одной прямой, перпендикулярной оси Ox .

Арю не может проезжать сквозь экраны, но должен обязательно проехать рядом с каким-либо из концов каждого экрана: там находятся энергетические станции, питающие экраны, от которых он планирует подзарядить свой энергоцикл. Помогите Арю посчитать, какое минимальное расстояние он должен проехать, чтобы добраться до космодрома.

Формат входа: В первой строке через пробел заданы два целых числа X — абсцисса точки, где расположен космодром — и n — количество защитных экранов. В следующих n строках описаны положения экранов. В i -й строке содержатся три целых числа x_i, y_i, y^i , где x_i — абсцисса концов i -го экрана, а y_i и y^i — ординаты. Известно, что $|y_i|, |y^i| \leq 1000$, $y_i < y^i$ и $0 < x_1 < x_2 < \dots < x_n < X$.

Формат выхода: Выдайте единственное вещественное число, которое не более, чем на 10^{-3} , отличается от истинного значения минимального расстояния, которое придется проехать Арю.

Пример

Вход: Выход:

10 3 18.176

1 -1 2

3 4 8

8 7 10

Система оценивания

<i>Подзадача</i>	<i>Баллы</i>	<i>Ограничения</i>	<i>Проверка</i>
1	15	$N \leq 2$	полная
2	35	$N \leq 18$	баллы
3	50	$N \leq 10^5$	баллы

9–11.5. «Отчет о проделанной работе». Уже знакомые нам Яна и Савов начали свой очередной рабочий день на складе: приступили к загрузке контейнера ящиками с микросхемами «Ореон» и макросхемами «Стук». И все бы ничего, но Йи Гроег (про которого мы также уже слышали) тоже не дремлет и время от времени, пока Савов и Яна отвлекаются на чашечку кофе, незаметно уносит один из ящиков с деталями.

Яна и Савов, конечно же, замечали пропажу ящиков с микросхемами, но ящики все равно отправлялись на утилизацию, так что наших героев не интересовала судьба груза. Что же их интересовало больше — так это отчеты о проделанной работе, которые они должны были предоставить в банк «Танакед». Отчет представляет собой суммарный вес ящиков, присутствовавших в контейнере на момент составления отчета и которые были погружены не ранее одного указанного момента и не позднее другого указанного момента.

Помогите Яне и Савову составить отчеты, если загрузка контейнера идет n минут, и в каждую минуту происходит ровно одно из описанных событий: Яна и Савов грузят в контейнер очередной ящик; Йи уносит ящик, который погрузили позже всех; Яна и Савов производят подсчет.

Формат входа: В первой строке дано число n — количество минут, в течение которого длилась загрузка контейнера. В следующих n строках перечисляются события: в i -ой строке описывается событие произошедшее в i -ую минуту. События бывают трех видов:

- 1) $1\ v$ — Яна и Савов погрузили очередной ящик массой v килограмм;
- 2) 2 — Йи унес ящик, который погрузили позже всех (конечно, если Йи приходит, когда контейнер пуст, он не забирает ничего);
- 3) $3\ t_1\ t_2$ ($0 \leq t_1 \leq t_2 < i$) — Яна и Савов считают суммарную массу всех ящиков с микросхемами, присутствующих в контейнере на текущий момент i и которые были погружены не ранее $(i - t_2)$ -й минуты и не позже $(i - t_1)$ -й минуты.

Присутствует хотя бы один запрос на составление отчета.

Формат выхода: Выведите ответы на все запросы в хронологическом порядке по одному в строке.

Пример 1

<u>Вход:</u>	<u>Выход:</u>
8	13
1 5	10
1 7	8
1 1	
1 2	
3 2 4	
3 2 4	
2	
3 4 6	

Система оценивания

<i>Подзадача</i>	<i>Баллы</i>	<i>Ограничения</i>	<i>Проверка</i>
1	10	$n \leq 10$, количество запросов ≤ 5	полная
2	20	$n \leq 100$, количество запросов ≤ 100	полная
3	35	$n \leq 3000$, количество запросов ≤ 10	баллы
4	35	$n \leq 10^5$, количество запросов $\leq 10^5$	баллы

Муниципальный этап Всероссийской олимпиады
школьников по информатике
в 2018 – 2019 учебном году
9–11 класс. Разбор решений и идеи тестов

9–11.1. «Выпить чашечку кофе...». Большинство подданных императора находятся на станции-цитадели Колизеум. В том числе, и Яна и Савов — среднестатистические подданные Империи, отнюдь не самого высокого звания, которые зарабатывают себе на жизнь разгрузкой и загрузкой контейнеров с микросхемами «Ореон» и макросхемами «Стук». Но даже простые грузчики Империи могут позволить себе раз в день напиток с десертником в одном из многочисленных кафе Колизеума. На каждом a -м этаже цитадели открыта кофейня «CofFix» (то есть на этажах $0, a, 2a, \dots$), а на каждом b -м — чайная «DifficulTea» (то есть на этажах $0, b, 2b, \dots$). Исторически сложилось так, что числа a и b взаимнопросты. Чтобы разнообразить свои трудовые будни, Яна и Савов решили как можно реже посещать один и тот же этаж. Помогите Яне и Савову посчитать, сколько дней они смогут ходить в питьевые заведения на разные этажи, прежде чем им придется посетить какой-то этаж снова.

Задача представляется программному комитету весьма простой. Формализованная постановка выглядит несложной: сколько в диапазоне от 0 до n имеется чисел, кратных либо a , либо b . Тонкость заключается в том, что числа, кратные и a , и b , должны учитываться только один раз.

Наивный алгоритм решения заключается в том, чтобы перебрать все числа из указанного диапазона, проверить каждое на делимость на a или на b , и посчитать количество делящихся. Однако такое решение уложится в ограничения по времени лишь для $n \leq 5 \cdot 10^7 \div 10^8$. Более разумная идея заключается в том, чтобы находить количество чисел, делящихся на a как $d_a = n \operatorname{div} a + 1$ (+1 для учета нуля). Соответственно, количество чисел, делящихся на b есть $d_b = n \operatorname{div} b + 1$, а делящихся и на a , и на b — $d_{ab} = n \operatorname{div} (a \cdot b) + 1$. Важно, что последняя формула верна только при взаимно простых числах a и b !

А дальше работает формула сложений-вычитаний: количество элементов в объединении двух множеств равно сумме элементов в этих множествах, уменьшенной на количество элементов в их пересечении,

$$|M_1 \cup M_2| = |M_1| + |M_2| - |M_1 \cap M_2|.$$

В нашем случае получается, что множество M_1 — это множество чисел, делящихся на a , и $|M_1| = d_a$, множество M_2 — множество чисел, делящихся на b , и $|M_2| = d_b$, $d_{ab} = |M_1 \cap M_2|$ — количество чисел, делящихся и на a , и на b . Отсюда находим искомую величину $|M_1 \cup M_2| = d_a + d_b - d_{ab}$.

Подзадача 1 может быть решена почти любым алгоритмом, подзадача 2 также решается прямым перебором. Для решения подзадачи 3 прямой перебор уже не подойдет.

9–11.2. «Саботаж». *Йи Гроег — обыкновенный, ничем не примечательный мальчик с планеты Рутнок, который, как и все юноши, мечтал о великих делах. По достижению совершеннолетия, Йи пошел в повстанческую Школу Подготовки Разведчиков, а после прохождения всех испытаний в ШПР Йи отправился работать на благо повстанцев и устроился на имперский склад на станции-цитадели Колизеум.*

Не прошло и недели с момента прибытия, как Йи Гроег раздобыл ценные планы имперцев по поставке деталей техники в дни с номерами от 1 до n . Особенно ценными для повстанцев являются микросхемы «Ореон» и макросхемы «Стук». Гроег знает, что в i -й день на склад привозят A_i микросхем стоимостью p ЗЕТов и B_i макросхем стоимостью q ЗЕТов. (Микросхемы стоят дороже макросхем.) Также он узнал, что в конце дня все схемы отправляются на завод, который тщательно охраняется, и бесполезно даже пытаться украсть детали оттуда.

За один день Йи может вынести со склада не больше S схем — ровно столько вмещается в бардачке его космолета. Какой наибольший ущерб Гроег может нанести Империи своими действиями и что ему для этого необходимо сделать?

Эта задача сделана общей для младшей и старшей групп, поскольку, с одной стороны, требует несложных размышлений о построении алгоритма, а с другой — имеет несложную техническую реализацию, допускающую реализацию, не привлекающую массивов. Собственно, основная идея решения — жадный алгоритм: каждый день нужно выносить как можно больше деталей первого типа, более дорогого (микросхем), а при наличии возможности забирать и сколько-то деталей второго типа (макросхем). Также нужно контролировать, что привезено деталей меньше, чем есть возможность забрать.

Подзадача 1 может быть решена и без написания цикла, подзадача 2 рассматривает ситуации, когда деталей меньше, чем может быть унесено. Подзадача 3 проверяет ситуации, когда имеются детали только первого вида. Подзадача 4 проверяет правильность полного алгоритма. Подзадача 5 дополнительно требует работы с 64-битными целыми типами.

9–11.3. «Праздник урожая Гречи». *Сегодня на планете Ксвежи, население которой известно своими кулинарными умениями, отмечают Праздник урожая Гречи. Ашас, уроженец этой планеты, хоть и находится вдали от родины, не хочет оставаться в стороне. Поэтому он решил приготовить праздничное блюдо, самое вкусное, какое только можно сделать из имеющихся у него n*

ингредиентов. Каждый ингредиент имеет регистрационный номер в Великой Поваренной книге Ксвежи — последовательность десятичных цифр, а его вкусность v равна сумме цифр его номера. Вкусность же всего блюда определяется как вкусностью используемых ингредиентов, так и последовательностью их добавления: вкусность блюда — это сумма разностей вкусностей соседних ингредиентов, где из вкусности предыдущего ингредиента вычитается вкусность следующего. Вкусность блюда может быть и отрицательной! Помогите Аша-су найти последовательность добавления ингредиентов, дающую самое вкусное блюдо.

Данная задача требует как определенной техники для своего решения, так и некоторых размышлений.

С технической точки зрения, для подсчета вкусности каждого ингредиента необходимо уметь перевести символ цифры в число, соответствующее этой цифре, чтобы затем просуммировать эти величины по всем символам. Для младших школьников эта задача может быть не вполне тривиальной.

После того, как числовой эквивалент каждой строки просчитан, нужно просчитать наилучшее их расположение. Однако здесь работает следующее несложное рассуждение. Пусть мы расположили строки в каком-то порядке, и их числовые эквиваленты есть $v_1, v_2, v_3, \dots, v_{n-2}, v_{n-1}, v_n$. Тогда числовой эквивалент такой последовательности равен

$$(v_1 - v_2) + (v_2 - v_3) + (v_3 - v_4) + \dots + (v_{n-2} - v_{n-1}) + (v_{n-1} - v_n) = v_1 - v_n.$$

Все слагаемые, кроме первого и последнего, взаимно уничтожились! Таким образом, для максимизации этой величины нужно максимизировать величину v_1 и минимизировать величину v_n ; значения и порядок остальных v_i неважны. То есть задача сводится к поиску минимума и максимума в массиве числовых эквивалентов отдельных строк. Строка, дающая максимум, выводится первой, дающая минимум — последней, остальные строки выводятся между ними в любом порядке, например, в том порядке, в котором они заданы во входном файле.

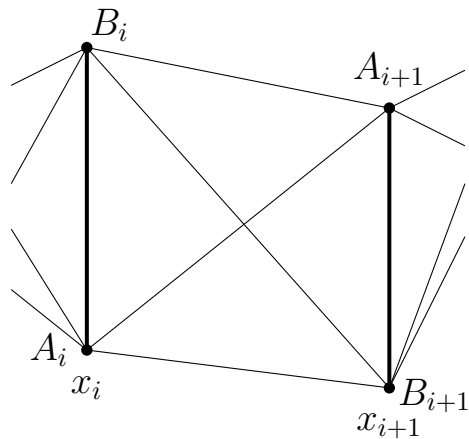
Подзадача 1, фактически, проверяет процедуру вычисления числового эквивалента строки. Подзадача 2 проверяет процедуру поиска оптимальной последовательности. Подзадача 3 осуществляет проверку программы в комплексе.

9–11.4. «Побег с планеты». *Сталкер Арю решил сбежать от повстанцев и стать частью Империи. Для этого ему необходимо добраться до космодрома планеты Рутнок. Он исследовал местность с помощью сканирующей прямой и обнаружил n защитных экранов, каждый из которых представляет собой отрезок. По данным сканирующей прямой Арю находится в точке с координатами $(0, 0)$, космодром — в точке $(X, 0)$, а каждый из защитных экранов перпендикулярен прямой между Арю и космодромом. При этом Арю знает, что никакие два экрана не расположены на одной прямой, перпендикулярной оси Ox .*

Арю не может проезжать сквозь экраны, но должен обязательно проехать рядом с каким-либо из концов каждого экрана: там находятся энергетические станции, питающие экраны, от которых он планирует подзарядить свой энергоцикл. Помогите Арю посчитать, какое минимальное расстояние он должен проехать, чтобы добраться до космодрома.

Данная задача представляется программному комитету имеющей сложность выше средней. В основном, сложность обуславливается разноплановостью идей, необходимых для ее решения. Решение данной задачи весьма сходно с решением задачи 7–8.5, поскольку формализация их постановок весьма близка.

Во-первых, необходимо привлекать геометрические размышления. Первая идея здесь — выгодно переходить последовательно от левых отрезков к правым в том порядке, в котором они заданы во входных данных: зигзаги вперед-назад лишь увеличат общее расстояние. Вторая идея — с любого из двух концов предыдущего отрезка можно перейти на любой из двух концов следующего отрезка, нет никаких препятствий между двумя соседними отрезками. Расстояние при этом вычисляется по известной формуле как длина отрезка с концами в указанных точках.



К решению задачи 9–11.4

Во-вторых, достаточно естественным образом возникает идея использования принципа динамического программирования. Пусть так или иначе мы имеем длины $d(A_i)$ и $d(B_i)$ кратчайших путей, ведущих из начальной точки в концы A_i и B_i отрезка с абсциссой x_i (см. рисунок). Тогда длина $d(A_{i+1})$ кратчайшего пути, ведущего в конец A_{i+1} отрезка с абсциссой x_{i+1} , есть

$$d(A_{i+1}) = \min \{d(A_i) + |A_i A_{i+1}|, d(B_i) + |B_i A_{i+1}|\}.$$

Аналогично, длина $d(B_{i+1})$ кратчайшего пути, ведущего в конец B_{i+1} отрезка с абсциссой x_{i+1} , есть

$$d(B_{i+1}) = \min \{d(A_i) + |A_i B_{i+1}|, d(B_i) + |B_i B_{i+1}|\}.$$

И тем самым, мы имеем длины $d(A_{i+1})$ и $d(B_{i+1})$ кратчайших путей, ведущих в концы A_{i+1} и B_{i+1} отрезка с абсциссой x_{i+1} .

То есть вычислительная процедура выглядит следующим образом. Имеем два массива A и B , содержащих длины кратчайших путей, ведущих в нижний и верхний концы очередного отрезка. Тогда $A[1] = |OA_1|$ и $B[1] = |OB_1|$, где $O(0, 0)$ — точка начального положения сталкера. Формулы пересчета:

$$\begin{aligned} A[i+1] &= \min \{A[i] + |A_i A_{i+1}|, B[i] + |B_i A_{i+1}|\}, \\ B[i+1] &= \min \{A[i] + |A_i B_{i+1}|, B[i] + |B_i B_{i+1}|\}. \end{aligned}$$

Ответ к задаче есть $\min \{A[n] + |A_n F|, B[n] + |B_n F|\}$, где $F(X, 0)$ — точка, где расположен космодром.

Заметим, что точность вычисления расстояний при указанных ограничениях на координаты точек не превосходит 10^{-12} : тип `double` хранит 18-19 десятичных цифр, из которых максимум 6 будет использовано под целую часть подкоренного выражения. Тем самым, суммарная ошибка при суммировании расстояний на максимально возможном количестве отрезков в 10^5 штук не превзойдет величины $10^5 \cdot 10^{-12} = 10^{-7}$, что с запасом укладывается в требования представления ответа.

В принципе, для решения этой задачи могут быть привлечены и графовые алгоритмы, например, алгоритм Дейкстры поиска кратчайшего пути в реберно-взвешенном графе. Однако он будет иметь бóльшую сложность $O(n \log n)$.

Подзадача 1 дает возможность получения ответа при любом алгоритме решения. Тесты в подзадаче 2 могут быть пройдены при реализации переборного алгоритма, когда для каждого отрезка поочередно перебираются каждый из его концов в качестве точки пути. Наконец, подзадача 3 нацелена на проверку оптимального алгоритма, работающего за линейное время.

9–11.5. «Отчет о проделанной работе». *Уже знакомые нам Яна и Савов начали свой очередной рабочий день на складе: приступили к загрузке контейнера ящиками с микросхемами «Ореон» и макросхемами «Стук». И все бы ничего, но Йи Гроег (про которого мы также уже слышали) тоже не дремлет и время от времени, пока Савов и Яна отвлекаются на чашечку кофе, незаметно уносит один из ящиков с деталями.*

Яна и Савов, конечно же, замечали пропажу ящиков с микросхемами, но ящики все равно отправлялись на утилизацию, так что наших героев не интересовала судьба груза. Что же их интересовало больше — так это отчеты о проделанной работе, которые они должны были предоставить в банк «Танакед». Отчет представляет собой суммарный вес ящиков, присутствовавших в контейнере на момент составления отчета и которые были погружены не ранее одного указанного момента и не позднее другого указанного момента.

Помогите Яне и Савову составить отчеты, если загрузка контейнера идет n минут, и в каждую минуту происходит ровно одно из описанных событий: Яна и Савов грузят в контейнер очередной ящик; Йи уносит ящик, который погрузили позже всех; Яна и Савов производят подсчет.

По мнению программного комитета, данная задача является весьма сложной, поскольку привлекает для своего решения достаточное несколько идей, связанных как с алгоритмами, так и со структурами данных, и которые непросты сами по себе.

Рассмотрим для начала ситуацию, когда ящики только укладываются в контейнер, но не извлекаются из него. Встает вопрос, как быстро посчитать суммарный вес ящиков, уложенных в контейнер в промежуток времени от минуты τ_1 до минуты τ_2 . Наивный подход, заключающийся в прямом поиске всех ящиков загруженных в контейнер в этот период времени и дальнейшем суммировании их весов, имеет сложность, пропорциональную размеру промежутка, то есть в худшем случае — $O(n)$. Замена линейного поиска бинарным не даст улучшения, так как тут самая долгая операция — именно суммирование. Известным подходом быстрого получения суммы чисел a_i , имеющих индексы из заданного интервала, является накопление массива S так называемых *частичных сумм*. А именно, вычисления по формулам $S[0] = 0$, $S[i+1] = S[i] + a_i$. То есть $S[i]$ — сумма всех чисел от первого до i -го. Тогда сумма чисел на отрезке от τ_1 до τ_2 включительно есть $S[\tau_2] - S[\tau_1 - 1]$.

Однако имеющаяся задача усложняется тем, что числа могут не только добавляться в набор, но и извлекаться из него. Можно следующим образом модифицировать идею массива частичных сумм для этой ситуации. А именно, заменим массив *частичных сумм* на стек пар

(момент времени, частичная сумма к этому моменту)

Соответственно, при добавлении ящика с весом v в контейнер в момент i мы считываем из пары (t, S) , находящейся на вершине стека, величину S и добавляем в стек пару $(i, S + v)$. При удалении из контейнера последнего загруженного туда ящика мы просто удаляем пару с вершины стека (если он непуст).

Тем самым, действительно, реализована идея хранения информации о сумме весов всех ящиков, находящихся в контейнере к определенному моменту. В отличие от массива последовательность моментов в парах уже может не быть непрерывной, поэтому для поиска суммы весов ящиков, загруженных в период с τ_1 до τ_2 , нужно искать элементы стека, в которых записан минимальный момент, не меньший τ_1 , и максимальный момент, не больший τ_2 , и вычитать из суммы во втором из найденных элементов сумму в первом из них. Поскольку элементы в стеке находятся в порядке увеличения моментов, то можно применять двоичный поиск. При этом, возможно, удобнее не использовать стандартный тип, реализующий стек, поскольку он почти наверняка не поддерживает возможность двоичного поиска, а самостоятельно реализовать стек на массиве, статическом или динамическом. Таким образом, сложность предложенной процедуры есть $O(m \log n)$, где m — количество запросов

Подзадача 1 является утешительной, ее тесты могут пройдены любым алгоритмом. Подзадача 2 также проходит практически любым алгоритмом, который

достаточно аккуратно реализован. Подзадача 3 может быть решена алгоритмом с прямолинейным подсчетом сумм весов ящиков в контейнере при условии аккуратного их хранения (на стеке). Наконец, подзадача 4 решается оптимальным алгоритмом, описанным выше.