

Технокубок 2017: Условия и разбор задач 2 отборочного раунда

А. Интервью с Олегом

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Поликарп взял у Олега интервью и записал его себе в блокнот без знаков препинания и пробелов, чтобы сэкономить время и успеть все записать. В итоге, интервью представляет собой строку s , состоящую из n строчных букв латинского алфавита.

В речи Олега есть слово-паразит `ogo`, а также все слова, которые получаются из слова `ogo` приписыванием справа к нему слога `go`. Например, слова `ogo`, `ogogo`, `ogogogo` являются паразитами, а слова `go`, `og`, `ogog`, `ogogog` и `oggo` — не являются.

Слова-паразиты имеют максимальный возможный размер, то есть, например, в речи `ogogoogoo` нельзя считать, что слово-паразит это `ogo`, а `goo` является частью обыкновенной фразы интервью. В данном случае словом-паразитом является подстрока `ogogo`.

До печати Поликарпу необходимо заменить каждое слово-паразит на последовательность из трёх звездочек. Обратите внимание, что независимо от длины слова-паразита оно заменяется ровно на три звёздочки.

Поликарп быстро справился с этой задачей. А сможете ли это сделать вы?

Время пошло!

Входные данные

В первой строке следует целое положительное число n ($1 \leq n \leq 100$) — длина интервью.

Во второй строке следует строка s длины n , состоящая из строчных букв латинского алфавита.

Выходные данные

Выведите текст интервью после замены каждого слова-паразита на «***».

Допустимо, что в ответе подстрока «***» будет идти подряд несколько раз.

Примеры

входные данные

7

aogogob

выходные данные

a***b

входные данные

13

ogogmgogogogo

выходные данные

gmg

входные данные

9

ogoogoogo

выходные данные

Примечание

В первом примере одно слово-паразит ogoго, поэтому интервью для печати выглядит как «a***b».

Во втором примере два слова-паразита `ogo` и `ogogogo`, поэтому интервью без слов-паразитов выглядит как «***gmg***».

Решение:

В этой задаче достаточно идти по строке слева направо и из каждого очередного индекса искать наидлиннейшую подстроку вида «`ogo...go`». Если такая найдена, то к ответу надо дописать «***» и перейти за её конец, иначе надо дописать к ответу очередную букву и перейти к следующей позиции.

В. Прожекторы

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Театральная сцена представляет собой прямоугольное поле размером $n \times m$. Директор театра выдал вам план сцены, согласно которому на ней будут располагаться актёры. На плане отмечено в каких клетках будут стоять актёры, а в каких нет.

Прожектор, установленный на сцену, будет светить в одном из четырёх направлений (если смотреть на план сцены сверху) — влево, вверх, вправо или вниз. Таким образом, под позицией прожектора понимается клетка, в которую он установлен, а также направление, в котором он светит.

Перед вами стоит задача поставить на сцену прожектор в *хорошую* позицию.

Позиция называется *хорошей*, если одновременно выполняются два условия:

- в соответствующей ей клетке нет актёра;
- в направлении, в котором светит прожектор, находится хотя бы один актёр.

Перед вами стоит задача посчитать количество *хороших* позиций для

установки прожектора. Две позиции установки прожектора считаются различными, если отличаются клетки расположения прожектора, или направление, в котором он светит.

Входные данные

В первой строке следует два целых положительных числа n и m ($1 \leq n, m \leq 1000$) — количество строк и количество столбцов в плане.

В следующих n строках следует по m целых чисел, каждое равно либо 0, либо 1, — описание плана. Если очередное число равно 1, то в соответствующей клетке находится актёр, а если 0, то клетка останется пустой. Гарантируется, что в плане есть хотя бы один актёр.

Выходные данные

Выведите единственное целое число — количество *хороших* позиций для установки прожектора.

Примеры

входные данные

```
2 4  
0 1 0 0  
1 0 1 0
```

выходные данные

```
9
```

входные данные

```
4 4  
0 0 0 0  
1 0 0 1  
0 1 1 0  
0 1 0 0
```

выходные данные

```
20
```

Примечание

В первом примере *хорошими* позициями для установки прожектора являются:

- клетка (1, 1) и направление вправо;
- клетка (1, 1) и направление вниз;
- клетка (1, 3) и направление влево;
- клетка (1, 3) и направление вниз;
- клетка (1, 4) и направление влево;
- клетка (2, 2) и направление влево;
- клетка (2, 2) и направление вверх;
- клетка (2, 2) и направление вправо;
- клетка (2, 4) и направление влево.

Таким образом, в данном примере всего 9 *хороших* позиций.

Решение:

Найдем количество *хороших* позиций, где прожектор направлен влево. Это можно сделать отдельно по каждой строке. Для этого надо сканировать строку слева направо, поддерживая флаг, что была встречена '1' (например, в переменной f). Тогда при обработке очередного значения:

- если оно равно '0', то к ответу следует прибавить единицу, если f равен `true`;
- если оно равно '1', то $f := \text{true}$.

Аналогично можно посчитать количество позиций для других трёх направлений.

С. Дорога до кинотеатра

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Вася находится в центре проката машин и хочет как можно скорее добраться до кинотеатра. Сеанс, на который он уже купил билет, начнётся через t минут. Считайте, что есть прямая дорога от центра проката машин до кинотеатра длиной s километров. Введём систему координат так, что центр проката машин находится в точке 0 , а кинотеатр находится в точке s .

Известно, что по пути от центра проката машин до кинотеатра есть k заправочных станций, причём на всех можно заливать неограниченное количество топлива совершенно бесплатно! Считайте, что операция заливки топлива осуществляется мгновенно.

В центре проката есть n машин, i -я из которых характеризуется двумя числами c_i и v_i — стоимостью аренды машины и вместимостью её бака. Таким образом, на заправке нельзя заливать в машину топлива больше вместимости её бака v_i . В центре проката все машины изначально полностью заправлены.

Каждая из машин может ехать в одном из двух скоростных режимов: обычном и ускоренном. В обычном режиме машина проезжает 1 километр за 2 минуты, при этом тратит на это 1 литр топлива. В ускоренном режиме машина проезжает 1 километр за 1 минуту, при этом тратит на это 2 литра топлива. Режим может быть изменён в любой момент. Изменять режим разрешается неограниченное количество раз.

Перед вами стоит задача выбрать машину с минимальной стоимостью аренды, на которой Вася успеет добраться до кинотеатра до начала своего сеанса, то есть не позднее, чем через t минут. Считайте, что в центре проката все машины изначально полностью заправлены.

Входные данные

В первой строке записаны четыре целых положительных числа n , k , s и t ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq 2 \cdot 10^5$, $2 \leq s \leq 109$, $1 \leq t \leq 2 \cdot 10^9$) — количество машин в центре проката, количество заправок на пути до кинотеатра, позиция кинотеатра и время, оставшееся до начала сеанса.

В следующих n строках следуют по два целых положительных числа c_i и v_i ($1 \leq c_i, v_i \leq 109$) — стоимость аренды i -й машины и объём её топливного бака.

В следующей строке следуют k **различных** целых чисел g_1, g_2, \dots, g_k ($1 \leq g_i \leq s - 1$) — позиции заправок на пути до кинотеатра в произвольном порядке.

Выходные данные

Выведите минимальную стоимость аренды подходящей машины, то есть такой, что Вася успеет добраться до кинотеатра до начала сеанса (не позднее, чем через t минут). Если ни одна из машин не подойдёт Васе, выведите -1.

Примеры

входные данные

3 1 8 10

10 8

5 7

11 9

3

выходные данные

10

входные данные

2 2 10 18

10 4

20 6

выходные данные

20

Примечание

В первом примере Вася успеет доехать до кинотеатра вовремя на первой и третьей машине, но выгоднее поехать на первой, стоимость аренды которой равна 10, а объём топливного бака равен 8. Тогда до первой заправки Вася сможет доехать в ускоренном режиме за 3 минуты, потратив на это 6 литров топлива, а затем, заправив полный бак, он сможет проехать 2 километра в обычном режиме за 4 минуты, потратив на это 2 литра бензина. Оставшиеся 3 километра он проедет в ускоренном режиме за 3 минуты, потратив на это 6 литров бензина.

Решение:

Понятно, что существует такое значение размера бака (назовём его w), что если машина имеет бак равный или больший w , то она доедет до кинотеатра вовремя, иначе — не успеет.

Значение w можно найти бинарным поиском, ведь функция $can(w)$ (сможет ли и успеет ли доехать машина) является монотонной — она сначала имеет значения `false`, затем `true`.

После нахождения w достаточно среди всех машин с размером бака w или более выбрать наиболее дешёвую.

Функцию $can(w)$ можно реализовать жадно моделируя процесс. Легко написать формулу для нахождения количество километров, которые можно проехать в режиме ускорения, если ближайшая заправка находится на расстоянии x , а сейчас у нас f литров бензина:

- если $x > f$, то доехать вообще нельзя и функция $can(w)$ должна вернуть `false`,
- если $x \leq f$, то в режиме ускорения проехать можно $\min(x, f - x)$ километров.

Таким образом, за один проход по массиву заправок в порядке возрастания их отдаления можно посчитать значение $cap(w)$.

D. Морской бой

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Галя играет в одномерный морской бой на поле размера $1 \times n$. В этой игре на клеточном поле расположены a кораблей, каждый состоит из b последовательных клеток. При этом одна клетка не может являться частью более чем одного корабля, однако, корабли **могут соприкасаться**.

Гале неизвестно положение кораблей. Галя может делать выстрелы по клеткам, при этом после каждого выстрела ей сообщается, является эта клетка частью какого-нибудь корабля (в таком случае считается, что Галя «попала»), или нет (тогда Галя «промахнулась»).

Галя уже сделала k выстрелов и все из них были промахами.

Перед вами стоит задача определить минимальное количество позиций, выстрелив в которые, Галя гарантированно попадет хотя бы в один из кораблей.

Гарантируется, что существует хотя бы одна расстановка кораблей, удовлетворяющая описанным условиям.

Входные данные

В первой строке находятся четыре целых положительных числа n, a, b, k ($1 \leq n \leq 2 \cdot 10^5, 1 \leq a, b \leq n, 0 \leq k \leq n - 1$) — длина поля, количество кораблей на поле, длина каждого корабля и количество выстрелов, которые Галя уже

произвела.

Во второй строке следует строка длины n , состоящая из нулей и единиц. Если i -й символ строки равен единице, то Галя уже стреляла в эту клетку, в противном случае, нет. Гарантируется, что в этой строке ровно k единиц.

Выходные данные

В первую строку выведите минимальное количество позиций, выстрелив в которые, Галя гарантированно попадёт хотя бы в один из кораблей.

Во вторую строку выведите позиции, в которые Галя должна стрелять.

Каждая позиция должна быть выведена ровно один раз. Позиции разрешается выводить в произвольном порядке. Позиции пронумерованы с 1 до n , начиная с самой левой.

Если существует несколько решений, выведите любое из них.

Примеры

входные данные

5 1 2 1

00100

выходные данные

2

4 2

входные данные

13 3 2 3

1000000010001

выходные данные

2

7 11

Примечание

В первом примере известно, что остался один корабль длины два. Он может располагаться как слева от уже сделанного выстрела (символа «1»), так и

справа. Таким образом, чтобы наверняка попасть в него, надо произвести два выстрела: один в левую часть, другой в правую.

Решение:

Заметим факт, что если на поле есть b подряд идущих нулей, то обязательно надо выстрелить в один из них. Предположим, что все корабли были максимально прижаты вправо. Поставим двойки в те клетки, где могут находиться максимально прижатые вправо корабли. Проитерируем по клеткам поля, начиная слева и будем стрелять в клетку, если в ней стоит 0 и до этого был $b - 1$ подряд идущий ноль. После этого останется выстрелить в одну любую клетку, в которой стоит двойка. Все описанные выстрелы и будут ответом.

Е. Подчинённые

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

В крупной компании работают n сотрудников, каждый из которых имеет уникальный номер от 1 до n . Из них **ровно** один сотрудник является главным, его номер равен s . Также известно, что все сотрудники, кроме главного, имеют ровно одного непосредственного начальника.

Каждому из сотрудников было поручено подать информацию о том, сколько начальников у него есть (не только непосредственных). Под начальниками сотрудника понимается его непосредственный начальник, а также непосредственный начальник непосредственного начальника данного

сотрудника, и так далее. Например, если в компании три сотрудника, первый из которых главный, у второго сотрудника непосредственный начальник — первый сотрудник, а у третьего сотрудника непосредственный начальник второй сотрудник, то у третьего сотрудника всего два начальника — один непосредственный и один не непосредственный. Главный сотрудник является начальником всех сотрудников, кроме самого себя.

Некоторые из сотрудников поторопились, ошиблись в подсчете и подали неверную информацию. Перед вами стоит задача определить минимально возможное число сотрудников, которые могли допустить ошибку при подаче информации.

Входные данные

В первой строке следует два целых положительных числа n и s ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq s \leq n$) — количество сотрудников и номер главного сотрудника.

Во второй строке следует n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq n - 1$), где a_i равно количеству начальников (не обязательно непосредственных) у сотрудника i , о котором он сообщил при подаче информации.

Выходные данные

Выведите минимально возможное число сотрудников, которые могли допустить ошибку при подаче информации.

Примеры

входные данные

3 2

2 0 2

выходные данные

1

входные данные

5 3

1 0 0 4 1

выходные данные

2

Примечание

В первом примере мог ошибиться, например, только первый сотрудник, в том случае, если:

- у первого сотрудника начальником является второй сотрудник,
- у третьего сотрудника начальником является первый сотрудник, второй сотрудник является главным.

Решение:

Изначально, если главный сотрудник сообщил, что у него есть начальники, заменим as на ноль. Если есть сотрудники, которые не являются главными, но сообщили число 0, будем считать, что они сообщили какое-нибудь число, большее, чем могли сообщить остальные сотрудники, например n .

Обязательно должен быть сотрудник, у которого ровно один начальник. Если такого нет, возьмем сотрудника, который сообщил максимальное число (учитывая всё описанное выше), и заменим это число на единицу. Аналогичную операцию нужно выполнить для числа 2, 3 и так далее, до тех пор, пока остаются сотрудники, которых мы еще не рассмотрели.

После того, как все сотрудники рассмотрены, осталось посчитать количество сотрудников, чьи числа были изменены — это и будет ответом.

Г. Игра финансистов

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Эта задача имеет нестандартное ограничение по памяти.

Финансистам Игорю и Жене стало скучно вечером, и они решили сыграть в игру. Для этого они приготовили n ценных бумаг, в которых содержится информация о доходе предприятия за какие-то промежутки времени. Обратите внимание, что доход может быть и положительным, и нулевым, и даже отрицательным.

Игорь и Женя выложили все бумаги в ряд и решили ходить по очереди. Игорь будет брать бумаги слева, а Женя справа. Первым ходит Игорь и берет 1 или 2 по своему выбору ценные бумаги слева. Далее, во время очередного хода игрок может взять k или $k + 1$ бумагу со своей стороны, если игрок, ходивший перед ним, взял ровно k бумаг. Ход пропускать не может ни один из игроков. Игра заканчивается, когда закончатся бумаги на столе, либо когда игрок не сможет сделать ход.

Перед вами стоит задача определить разность между суммой доходов бумаг, которые забрал себе Игорь, и суммой доходов бумаг, которые забрал себе Женя, если оба игрока играют оптимально. Игорь хочет максимизировать разность, а Женя — минимизировать.

Входные данные

В первой строке следует целое положительное число n ($1 \leq n \leq 4000$) — количество бумаг.

Во второй строке следует последовательность из n целых чисел a_1, a_2, \dots, a_n ($-105 \leq a_i \leq 105$), где a_i равно доходу, информация о котором содержится на i -й бумаге.

Выходные данные

Выведите разность между суммой доходов бумаг, которые забрал себе Игорь, и суммой доходов бумаг, которые забрал себе Женя, если оба игрока играют оптимально. Игорь хочет максимизировать разность, а Женя —

минимизировать.

Примеры

входные данные

3

1 3 1

выходные данные

4

входные данные

5

-1 -2 -1 -2 -1

выходные данные

0

входные данные

4

-4 -2 4 5

выходные данные

-13

Примечание

В первом примере Игорю выгодно забрать две бумаги слева, получив суммарный доход 4, тогда Женя не сможет сделать ход, так как осталась всего одна бумага, а он может взять либо 2, либо 3 на текущем ходе.

Решение:

Будем решать задачу методом динамического программирования. Достаточно понятно, что позиция характеризуется тремя числами: границами отрезка бумаг, которые все еще лежат на столе, и количеством бумаг, которые взял предыдущий игрок; а также очередностью хода. Поэтому, пусть $llrk$ это результат игры, если бы на столе изначально лежали только бумаги с l

по r , первым ходил Игорь, и делал бы ход на k или $k + 1$. Аналогично, пусть Z_{lrk} — то же, но первым ходит Женя. Ясно, что в общем случае

$$I_{lrk} = \max(Z_{l+k,r,k} + \sum_{i=l}^{i<l+k} a_i, Z_{l+k+1,r,k+1} + \sum_{i=l}^{i<l+k+1} a_i),$$

$$Z_{lrk} = \min(I_{l,r-k,k} - \sum_{i=r-k+1}^{i\leq r} a_i, I_{l,r-k-1,k+1} - \sum_{i=r-k}^{i\leq r} a_i).$$

Надо аккуратно обработать случаи, когда игрок не может забрать нужное число бумаг. Ответ на задачу — значение I_{1n1} .

На первый взгляд кажется, что такое решение имеет асимптотику $O(n^3)$. Однако при пристальном рассмотрении это не так. Какие значения могут принимать l , r и k ?

Во первых, $\frac{k(k+1)}{2} \leq n$, т. к. если последний игрок взял k бумаг, то всего взято уже не менее $1 + 2 + 3 + \dots + k = \frac{k(k+1)}{2}$ бумаг. Отсюда, k не превышает $\sqrt{2n}$.

Во вторых посмотрим на разность числа бумаг, взятых Женей и Игорем, то есть на величину $d = (n - r) - (l - 1)$. Пусть при этом игроки сделали поровну ходов, то есть сейчас ходит Игорь. Тогда $0 \leq d \leq k - 1$. Действительно, на каждом ходу Женя берет либо столько же бумаг, сколько и Игорь, либо на одну больше, при этом увеличивается «длина» хода. Всего длина хода увеличилась на $k - 1$, а значит, эта разность не больше $k - 1$. Таким образом, мы можем нумеровать состояния числами l , d и k , при этом всего состояний $O(n^2)$. Состояния, в которых ход Жени, не будем рассматривать, а сразу добавим в переход и перебор обоих возможных ответных ходов (всего четыре перехода). Итоговая асимптотика $O(n^2)$, при этом проще всего реализовать данное решение с помощью рекурсивного перебора с запоминанием.