

Общая информация по задачам первого тура

Задача	Тип задачи	Ограничения
1. Лестница для участников олимпиады	стандартная	0.5 с, 1024 МБ
2. Пересменка в Сириусе	стандартная	1 с, 1024 МБ
3. Сочи Парк	стандартная	1 с, 1024 МБ
4. Лягушки на дереве	стандартная	2 с, 1024 МБ

Необходимо считывать данные из стандартного потока ввода. Выходные данные необходимо выводить в стандартный поток вывода.

Баллы за подзадачу начисляются только если все тесты этой и необходимых подзадач пройдены. Решение запускается на тестах для определенной подзадачи, если все тесты всех необходимых подзадач пройдены. В одной из задач можно получить частичные баллы за подзадачу. Для тестирования подзадачи достаточно, чтобы во всех необходимых подзадачах был получен положительный балл.

Во всех подзадачах каждой задачи во время тура вам показываются баллы за подзадачу, если все тесты пройдены, либо первая ошибка и номер теста.

Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач указана дополнительно буква У.

Задача 1. Лестница для участников олимпиады

Ограничение по времени: 0.5 секунд
Ограничение по памяти: 1024 мегабайта

В ОЦ «Сириус» любимым местом для сбора и неформального общения школьников служат различные лестницы. Но количество участников олимпиады по информатике значительно превосходит количество участников любой образовательной программы, и подходящей для них лестницы среди имеющихся не нашлось, поэтому служба оснащения решила построить новую лестницу, используя специальную заготовку.

Заготовка представляет собой таблицу из h строк и w столбцов, пронумерованных сверху вниз и слева направо соответственно. В каждой клетке таблицы записано одно число — ноль или единица. *Лестницу* можно сделать только из тех клеток таблицы, в которых записана единица.

Полученная лестница образуется из множества клеток, в которых записана единица, находящихся в нескольких последовательных строках таблицы. Множество выбранных клеток в каждой строке лестницы должно быть непрерывным отрезком. При этом в каждой следующей строке, входящей в лестницу, должно быть выбрано не меньше клеток, чем в предыдущей, находящейся непосредственно над ней, строке, а самые левые выбранные клетки в каждой строке должны располагаться в одном и том же столбце.

Ниже приведен пример лестницы.

1		
1	1	
1	1	
1	1	1

Найдите в заданной таблице максимальное количество клеток, образующих лестницу.

Формат входных данных

Первая строка входных данных содержит два целых числа h и w ($1 \leq h, w \leq 2 \cdot 10^5$, $h \cdot w \leq 4 \cdot 10^6$) — количество строк и столбцов таблицы соответственно.

Каждая из следующих h строк содержит по w символов, каждый из которых равен 0 или 1 — числа, написанные в клетках таблицы.

Формат выходных данных

Выведите одно число — максимальное количество клеток, образующих лестницу.

Система оценивания

Подзадача	Баллы	Ограничения	Необходимые подзадачи
		h, w	
1	25	$h, w \leq 50$	У
2	25	$h, w \leq 400$	У, 1
3	25	$h \cdot w \leq 200\,000$	У, 1, 2
4	25	—	У, 1–3



Пример

стандартный ввод	стандартный вывод
6 4 0011 1101 0111 1110 0111 0100	8

Замечание

Ниже изображен рисунок для первого примера. Лестница, состоящая из максимально возможного количества клеток таблицы, отмечена серым цветом.

0	0	1	1
1	1	0	1
0	1	1	1
1	1	1	0
0	1	1	1
0	1	0	0

Задача 2. Пересменка в Сириусе

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Участники образовательных программ иногда задумываются, почему между двумя программами обычно бывает перерыв в несколько дней. Ответ прост: сотрудникам Сириуса необходимо после очередной программы привести в порядок жилые номера.

На одном этаже в гостинице ОЦ «Сириус» находятся n номеров, пронумерованных от 1 до n . После проведения образовательной программы все эти номера нуждаются в ремонте.

К ремонтным работам привлечены k сотрудников, пронумерованных от 1 до k . За i -м сотрудником закреплён диапазон номеров с l_i по r_i включительно, а также зафиксирован номер m_i из этого диапазона, с которого он должен начать обход своих номеров. Диапазоны номеров у разных сотрудников могут пересекаться и даже совпадать.

Сотрудники в некотором порядке направляются с базы для выполнения работ. Следующий сотрудник направляется только после возвращения предыдущего на базу.

Когда i -го сотрудника направляют на выполнение работ, он сначала идёт в номер m_i . Если этот номер всё ещё нуждается в ремонте, то сотрудник ремонтирует его, а также посещает все номера из диапазона с l_i по r_i , за который он отвечает, и ремонтирует все нуждающиеся в ремонте номера из этого диапазона, после чего возвращается на базу. После этого все номера из диапазона с l_i по r_i более не нуждаются в ремонте.

Если же первый посещённый сотрудником номер m_i не нуждается в ремонте, поскольку его уже отремонтировали ранее направленные для выполнения работ коллеги, то сотрудник сразу возвращается на базу, надеясь, что коллеги уже отремонтировали и все остальные номера из его диапазона. В этом случае некоторые другие номера из диапазона с l_i по r_i всё ещё могут нуждаться в ремонте.

Определите, можно ли при подобном подходе сотрудников к выполнению своих обязанностей направить их всех для выполнения работ в таком порядке, чтобы в итоге все номера от 1 до n оказались отремонтированы.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и k ($1 \leq n, k \leq 5 \cdot 10^5$) — количество номеров и количество сотрудников соответственно.

В каждой из последующих k строк содержится три целых числа l_i , m_i и r_i ($1 \leq l_i \leq m_i \leq r_i \leq n$) — первый номер диапазона ответственности i -го сотрудника, номер из диапазона, с которого он должен начать обход своих, и последний номер из его диапазона, соответственно.

Гарантируется, что сумма n и k по всем наборам входных данных не превосходит $5 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите «YES», если можно отремонтировать все номера, и «NO» — в противном случае.



Система оценивания

Обозначим за N сумму n по всем наборам входных данных, за K — сумму k по всем наборам входных данных.

Подзадача	Баллы	Дополнительные ограничения			Необх. подзадачи
		n, N	k, K	дополнительно	
1	5	—	$K \leq 10\,000$	$m_i = l_i$	
2	5	$N \leq 500$	$k \leq 8$		У
3	2	$n \leq 18$	$K \leq 500$		У
4	12	$n \leq 50$	$K \leq 50$		У
5	9	$n \leq 150$	$K \leq 150$		У, 4
6	8	$N \leq 500$	$K \leq 500$		У
7	6	—	$K \leq 10\,000$	За каждым сотрудником закреплен номер 1 или номер n	—
8	18	—	$K \leq 10\,000$	Для каждого сотрудника найдется номер, который закреплён только за ним	—
9	3	—	—	Для каждого сотрудника найдется номер, который закреплён только за ним	8
10	4	—	$K \leq 10\,000$	$r_i - l_i = r_j - l_j$ для любых i, j	—
11	4	—	$K \leq 10\,000$	Любое m_i совпадает с l_i или r_i	1
12	4	$n \leq 10\,000$	$K \leq 10\,000$	—	У, 2–6
13	6	—	$K \leq 10\,000$		У, 1–8, 10–12
14	14	—	—		У, 1–13

Пример

стандартный ввод	стандартный вывод
2	YES
5 2	NO
3 4 5	
1 3 3	
5 3	
1 2 4	
2 4 5	
3 3 3	

Замечание

В первом наборе входных данных из примера нужно сначала направить для выполнения ремонтных работ второго сотрудника, он отремонтирует номера с первого по третий. Затем первый сотрудник направится в номер 4. Так как он еще нуждается в ремонте, первый сотрудник отремонтирует оставшиеся номера в своем диапазоне. В результате все номера будут отремонтированы.

Во втором наборе данных выбрать подходящий порядок отправки сотрудников невозможно.

Задача 3. Сочи Парк

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта



В Сочи Парке открылся новый аттракцион. Вдоль прямой расположены n целей, координата i -й цели равна x_i ($1 \leq i \leq n$). Посетители должны поразить все эти цели в произвольном порядке. Для поражения целей используются мячики. Если посетитель находится в точке с координатой x и хочет поразить цель, находящуюся в точке x_i , ему потребуется потратить $(x - x_i)^2$ калорий.

Посетитель входит в аттракцион в точке с координатой x_0 . Неограниченные запасы мячиков находятся в точке входа, а также во всех точках на расстоянии d друг от друга, то есть в точках $x_0 + kd$, где k — произвольное целое число. Переносить мячики запрещено правилами аттракциона, поэтому бросать их можно только из этих точек.

В день между турами m участников олимпиады посетят Сочи Парк. Участники соревнования находятся в разной физической форме, поэтому j -му участнику олимпиады для перемещения на расстояние d требуется t_j калорий.

Вам нужно определить, какое минимальное число калорий необходимо каждому участнику для поражения всех целей аттракциона.

Формат входных данных

В первой строке задано одно целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество целей в аттракционе. Во второй строке заданы n целых чисел $x_1, x_2, \dots, x_n$ ($0 \leq x_i \leq 10^9$) — координаты целей.

В третьей строке заданы два целых числа x_0 и d ($0 \leq x_0 \leq 10^9$, $1 \leq d \leq 2 \cdot 10^6$) — точка входа посетителя аттракциона и расстояние между местами нахождения запасов мячиков.

В четвертой строке задано одно целое число m ($1 \leq m \leq 6 \cdot 10^5$) — количество участников олимпиады.

В следующих m строках содержится по одному целому числу t_j ($0 \leq t_j \leq 10^8$) — количество энергии, необходимое j -му участнику олимпиады для перемещения между двумя соседними местами нахождения запасов мячиков.

Формат выходных данных

Для каждого участника олимпиады выведите одно целое число — минимальное количество, необходимое ему для перемещения и поражения всех целей.

При данных ограничениях ответ не превосходит максимального значения 64-битного знакового типа данных. Однако для промежуточных вычислений может понадобиться тип данных `__int128` в C++ (поддерживается только в компиляторе GNU C++), `BigInteger` в Java, `int` в Python.

Система оценивания

Подзадача	Баллы	Доп. ограничения				Необх. подзадачи
		n	x_0	m	дополнительно	
1	9	–	–	$m = 1$	$t_1 = 0$	–
2	7	$n = 1$	–	$m \leq 10\,000$	–	–
3	8	$n = 2$	–	$m \leq 10\,000$	$x_1 \leq x_0 \leq x_2$	–
4	3	$n \leq 50$	$x_0 = 0$	$m \leq 50$	$d \leq 50, x_i \leq 50$	–
5	2	$n \leq 50$	$x_0 \leq 50$	$m \leq 50$	$d \leq 50, x_i \leq 50$	4
6	4	–	$x_0 = 0$	$m \leq 10$	$x_i \leq 10^6$	–
7	2	–	$x_0 \leq 10^6$	$m \leq 10$	$x_i \leq 10^6$	У, 6
8	6	–	$x_0 = 0$	$m \leq 10\,000$	$x_i \leq 10^6$	4, 6
9	10	–	$x_0 \leq 10^6$	$m \leq 10\,000$	$x_i \leq 10^6$	У, 4–8
10	7	–	$x_0 \leq 10^6$	$m \leq 10^5$	$x_i \leq 10^6$	У, 4–9
11	2	–	–	$m \leq 10$	–	У, 1, 6, 7
12	12	–	$x_0 = 0$	$m \leq 10^5$	$d = 1$	–
13	5	–	–	$m \leq 10^5$	$d = 1$	12
14	8	–	$x_0 = 0$	$m \leq 10^5$	–	4, 6, 8, 12
15	2	–	–	$m \leq 10^5$	–	У, 1–14
16	1	–	–	$m \leq 2 \cdot 10^5$	–	У, 1–15
17	3	–	–	$m \leq 3 \cdot 10^5$	–	У, 1–16
18	9	–	–	–	–	У, 1–17

Примеры

стандартный ввод	стандартный вывод
3 4 0 7 2 3 7 0 1 2 3 4 23 25	3 7 10 12 13 32 33
4 30 239 57 179 0 7 5 1 10 15 100 100000	49 355 525 3378 93311
4 100 2 101 666 9 10 5 777 1 2 15 10	49597 91 159 1043 703

Замечание

В первом тесте для второго участника ($t_2 = 1$) оптимальным будет следующий алгоритм поражения целей:

1. Переместиться из точки $x_0 = 2$ в точку $x_0 - d = -1$, потратив $t_2 = 1$ калорию. Обратите внимание, координата посетителя может быть отрицательной.
2. Поразить цель в точке $x_2 = 0$, потратив $(-1 - 0)^2 = 1$ калорию.
3. Переместиться в точку $-1 + 2d = 5$, потратив $2t_2 = 2$ калории.
4. Поразить цель в точке $x_1 = 4$, потратив $(5 - 4)^2 = 1$ калорию.
5. Переместиться в точку $5 + d = 8$, потратив $t_2 = 1$ калорию.
6. Поразить цель в точке $x_3 = 7$, потратив $(8 - 7)^2 = 1$ калорию.

Суммарные затраты энергии равны $1 + 2 + 1 + 1 + 1 + 1 = 7$ калорий. Можно показать, что это минимальное количество энергии.

Для шестого участника ($t_6 = 23$) оптимальным будет следующий алгоритм поражения целей:

1. Поразить цель в точке $x_2 = 0$, потратив $(2 - 0)^2 = 4$ калории.
2. Переместиться в точку $2 + d = 5$, потратив $t_6 = 23$ калории.
3. Поразить цель в точке $x_3 = 7$, потратив $(7 - 5)^2 = 4$ калории.
4. Поразить цель в точке $x_1 = 4$, потратив $(5 - 4)^2 = 1$ калорию.

Суммарные затраты энергии равны $4 + 23 + 4 + 1 = 32$ калории. Можно показать, что это минимальное количество энергии.

Задача 4. Лягушки на дереве

Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

На ФТ Сириус можно наблюдать не только обыкновенных, но и древесных лягушек, про некоторые виды которых известно, что они могут менять свой цвет с зелёного на коричневый, и наоборот.

Как известно, *дерево* — это связный граф без циклов. В каждой вершине дерева живёт ровно одна лягушка. Изначально все лягушки имеют зелёный цвет. Лягушки могут прыгать по дереву. За один прыжок лягушка перемещается из вершины дерева, в которой она находится, в соседнюю с ней по ребру вершину. После каждого прыжка цвет лягушки меняется на противоположный.

Лягушки любят петь дуэтом. Дуэт обязательно должен состоять из двух лягушек разного цвета. Чтобы две лягушки образовали дуэт, одна из лягушек должна добраться до вершины, где живет другая лягушка, совершив при этом не более d прыжков. Чтобы после перемещения цвет гостя отличался от цвета хозяйки, гостя должна сделать нечётное количество прыжков.

Необходимо определить, какое максимальное количество дуэтов лягушек может образоваться. Каждая лягушка может входить только в один дуэт. Если вы правильно определите максимальное количество дуэтов, вы получите частичный балл за подзадачу. Чтобы получить полный балл за подзадачу необходимо также выяснить, какие пары лягушек должны образовать дуэты, чтобы их оказалось максимальное количество.



Формат входных данных

Первая строка входных данных содержит одно целое число n ($2 \leq n \leq 5 \cdot 10^5$) — количество вершин в дереве.

Вторая строка входных данных содержит одно целое нечётное число d ($1 \leq d \leq n - 1$) — максимальное количество прыжков, которое может сделать одна лягушка на пути к другой.

Каждая из следующих $n - 1$ строк входных данных содержит два целых числа u и v ($1 \leq u, v \leq n$) — номера вершин дерева, соединённых одним ребром. Вершины пронумерованы от 1 до n .

Формат выходных данных

В первой строке выведите одно целое число m — максимально возможное количество дуэтов лягушек, которые могут образоваться.

Если вы не хотите предъявлять сами пары, то выведите в следующей строке число -1 и завершите работу программы.

Иначе, в следующих m строках выведите по два целых числа u_i и v_i — пару вершин, лягушки из которых должны встретиться в одной из этих вершин и образовать дуэт, соблюдая описанные выше правила.

Если максимальное количество дуэтов может быть образовано несколькими способами, выведите любой из них.

Система оценивания

Если решение выводит не максимальное возможное количество пар или некорректный набор пар на одном из тестов подзадачи, то оно получает 0 баллов за подзадачу. Если хотя бы на одном тесте подзадачи решение выводит -1 вместо набора пар и на каждом тесте подзадачи выводит либо верный набор пар, либо -1 , то оно получает половину баллов за подзадачу. Если на каждом тесте подзадачи решение выводит верный набор пар, то оно получает полный балл за подзадачу.

Подзадача	Баллы	Дополнительные ограничения		Необх. подзадачи
		n	d	
1	6	$n \leq 14$	–	У
2	6	$n \leq 300\,000$	$d = n - 1$	–
3	10	$n \leq 300\,000$	$d = 1$	–
4	14	$n \leq 300\,000$	$d = 3$	–
5	8	$n \leq 200$	–	У, 1
6	12	$n \leq 30\,000$	$d \leq 9$	У
7	4	$n \leq 300\,000$	$d \leq 13$	У, 1, 3, 4, 6
8	10	$n \leq 300\,000$	$d \leq 99$	У, 1, 3, 4, 6, 7
9	14	$n \leq 300\,000$	–	У, 1–8
10	16	–	–	У, 1–9

Примеры

стандартный ввод	стандартный вывод
8 7 1 2 2 3 3 4 3 5 1 6 6 7 3 8	3 2 7 6 3 4 1
11 3 1 2 2 3 3 4 3 5 3 6 3 7 1 8 8 9 8 10 8 11	4 3 7 11 8 10 2 1 6