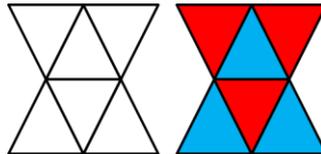


ОПРЕДЕЛЯЕМ, ЧТО ЯВЛЯЕТСЯ ОТВЕТОМ

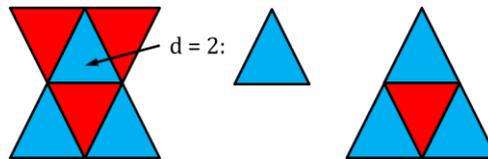
Прежде всего следует понять, что в задаче требуется подсчитать количество различных треугольников, входящих в состав фигуры.



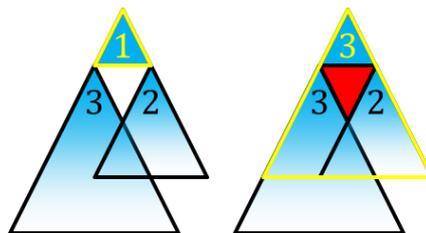
Чтобы упростить дальнейшие рассуждения, разделим элементарные треугольники в фигуре на «синие», направленные вверх, и «красные», направленные вниз:



Как найти ответ на задачу (пока не обращаясь к языку программирования)? Воспользуемся идеей динамического программирования. Для каждого синего треугольника T посчитаем $d[T]$ – число различных треугольников, самым верхним элементом которых является синий треугольник T .



Немного подумав, обнаруживаем, что $d[T]$ также можно определить как размер самого большого треугольника, самым верхним элементом которого является синий треугольник T . Эту величину достаточно просто пересчитывать. Если под T нет красного треугольника, то $d[T] = 1$, иначе $d[T] = 1 + \min(d[T_A], d[T_B])$, где T_A и T_B – синие треугольники под T .

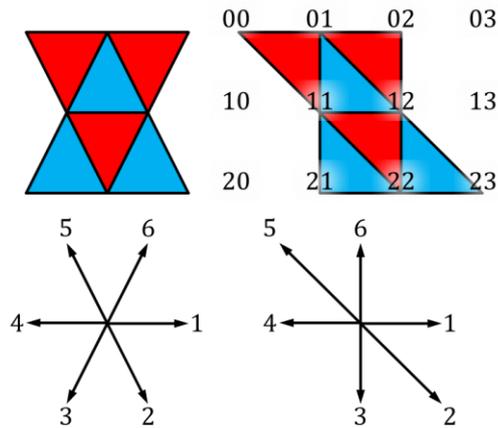


Для красных треугольников расчёты производятся аналогично.

Ответом на задачу является сумма $d[T]$ для всех синих и красных треугольников.

ОБРАБАТЫВАЕМ ВВОД

Чтобы хоть как-то отождествить треугольники в фигуре и ячейки массива, изменим вид треугольников и направления перемещений при обходе:



Будем считать, что каждый треугольник отождествляется с ячейкой, координаты которой задаются координатами левой верхней точки треугольника (следует обратить внимание, что ячейки массива индексируются с нуля, а координаты вершин треугольника могут быть отрицательными; следует ввести смещение).

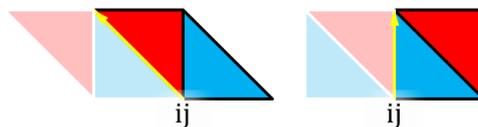
Так, синие треугольники на иллюстрации соответствуют ячейкам $(0; 1)$, $(1; 1)$ и $(1; 2)$, а красные – ячейкам $(0; 0)$, $(0; 1)$ и $(1; 1)$.

При считывании ввода мы должны для каждой строки i (соответствующей некоторой координате y) определить, где в этой строке начинаются и заканчиваются области синих и красных треугольников.

Крайне важно понять, что при обходе по часовой стрелке перемещения #2 и #3 будут определять концы полосок из треугольников, а перемещения #5 и #6 – начала таких полосок.



При перемещении из ячейки $(i; j)$ в направлении #2 область синих треугольников прекращается с позиции $(i; j + 1)$, область красных треугольников прекращается с позиции $(i; j)$. При перемещении в направлении #3 области синих и красных треугольников прекращаются с позиции $(i; j)$.



При перемещении из ячейки $(i; j)$ в направлении #5 область синих треугольников начинается с позиции $(i - 1; j)$, область красных треугольников начинается с позиции $(i - 1; j - 1)$. При перемещении в направлении #6 области синих и красных треугольников начинаются с позиции $(i - 1; j)$.

Таким образом, при чтении ввода для каждой координаты i мы устанавливаем, в каких отрезках по координате j присутствуют синие и красные треугольники.

ИССЛЕДУЕМ ОРИГИНАЛЬНОЕ РЕШЕНИЕ

Достаточно нетрудно найти в интернете [архив материалов рассматриваемой задачи](#). Он содержит единственное решение triangle_pm.dpr (pm – Пётр Митричев?)



После обработки ввода данное решение просто заполняет два двумерных массива $10^4 \times 10^4$ и по ним считает два таких же массива со значениями динамики. Но погодите-ка...



На этом закончим исследование оригинального решения и приступим к оптимизации собственного.

Мы не можем явно отметить в двумерном массиве все ячейки, в которых находятся треугольники, поэтому продолжаем хранить только начала и концы их полосок (например, в `tar'ax`).

Далее, можно заметить, что для пересчёта динамики требуется хранить только две строки – текущую и предыдущую. При этом для красных треугольников динамика остаётся без изменений, а для синих её придётся модифицировать: сохранять максимальные значения не в верхнем элементарном треугольнике, а в нижнем правом. Так мы сможем пересчитывать значения динамики и для красных, и для синих треугольников, храня только две строки и двигаясь по второй слева направо. (Обратите внимание, что при этом критично сначала рассчитывать динамику для красных треугольников, потом для синих).

Кратко повторим последовательность действий:

- Пока есть нерассмотренные участки с треугольниками
 - Скопировать строку #1 в строку #0, заполнить строку #1 нулями
 - Пока есть участки, относящиеся к текущей координате y
 - Отметить в строке #1 все позиции треугольников, для каждой из них посчитать значение d
 - Суммировать все получаемые значения d
 - Перейти к следующей координате y

Похоже, по-настоящему геморройная реализация только начинается.

ТЕСТИРУЕМ

[Решение, которое должно укладываться в ML](#), готово. Воспользуемся тестами из архива материалов, чтобы проверить правильность решения.

В нашем распоряжении 10 тестов – семпл, шестиугольник со стороной 1, макстест, случайный тест и 6 одинаковых треугольников со стороной 1, заданных разными способами. Скорее всего, интерес представляют только два последних теста.

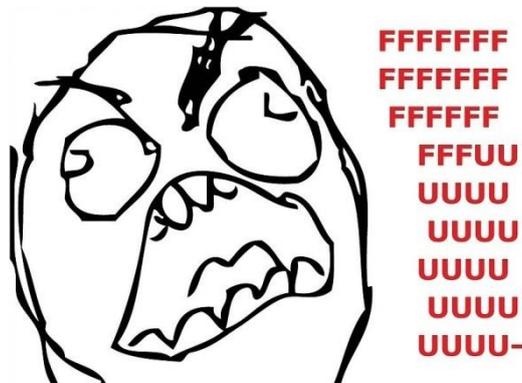
Обнаруживаем, что составленное решение выдаёт ответы, совпадающие с авторскими, на всех тестах, кроме #9 (макстест).

Тщательно тестируем решение и ничего не находим.

Запускаем авторское решение a ideone и получаем на макстесте новый, третий вариант ответа.

Отправляем собственное решение в тестирующую систему.

Получаем Accepted.



Кстати, на олимпиаде эту задачу (G) за отведённое время успели решить две команды школьников.

Результаты

neerc.ifmo.ru/school/io/archive/20051119/standings-20051119-advanced.html

Цикл интернет-олимпиад. Третья олимпиада, усложненный уровень

Попыток: 823, удачных: 86
Последняя удачная попытка: Tasha&Gosha.[1543] (Москва, Россия), E, 4:59:59

Команда	A	B	C	D	E	F	G	H	I	=	Время
KGS (Новосибирск, Россия)	+2 204:19	+2 168:03	.	+2 63:05	+5 183:54	+ 36:18	+2 282:32	.	+ 227:59	7	1423
Бурундучки (Санкт-Петербург)	+ 174:53	-4	-6	+1 32:55	+1 191:04	+1 70:50	+3 282:05	.	+12 199:27	6	1308
Unknown_Lamers (Екатеринбург)	+ 257:01	+ 24:30	.	+ 44:05	+ 125:02	+ 82:13	-8	.	.	5	532
next_permutation(a.begin(), a.end()) (Москва)	+ 267:32	-6	-8	+ 74:13	+2 286:32	+2 151:57	.	.	-4	4	858
LPN_3 (Саратов, Россия)	.	-1	.	+ 53:22	+1 253:35	+2 137:16	.	.	+6 276:42	4	899
SG1 (г. Салават, Россия)	-11	-1	.	+3 111:30	+ 130:07	+ 43:23	.	.	-6	3	344