

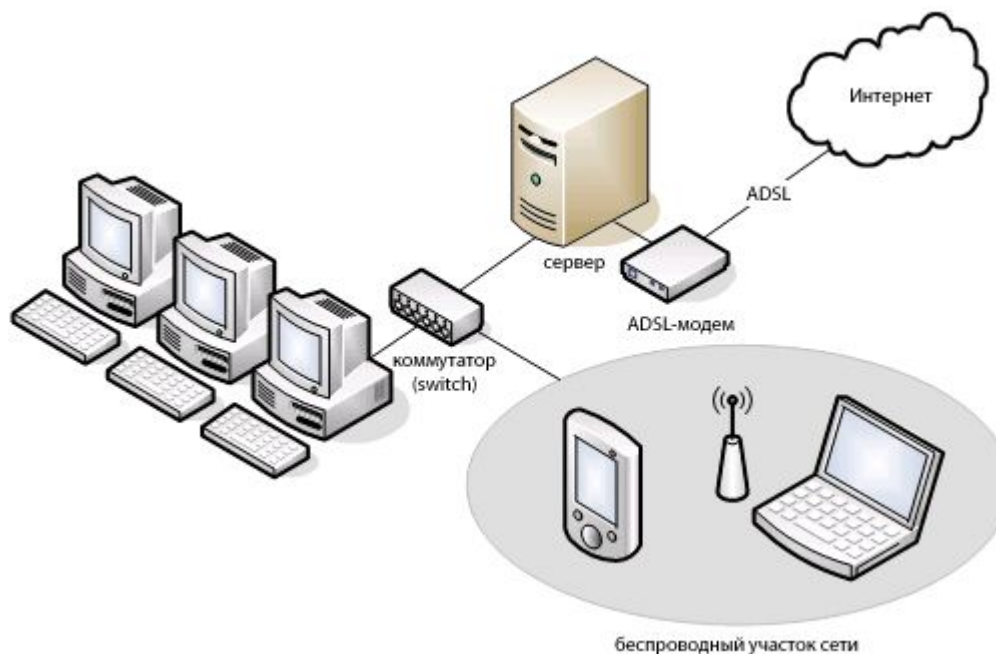
XXI Всероссийская олимпиада школьников по информатике

Разбор задач
7 апреля 2009 года

Новосибирск



Задача 1. Компьютерная сеть



Автор задачи – Сергей Копелиович

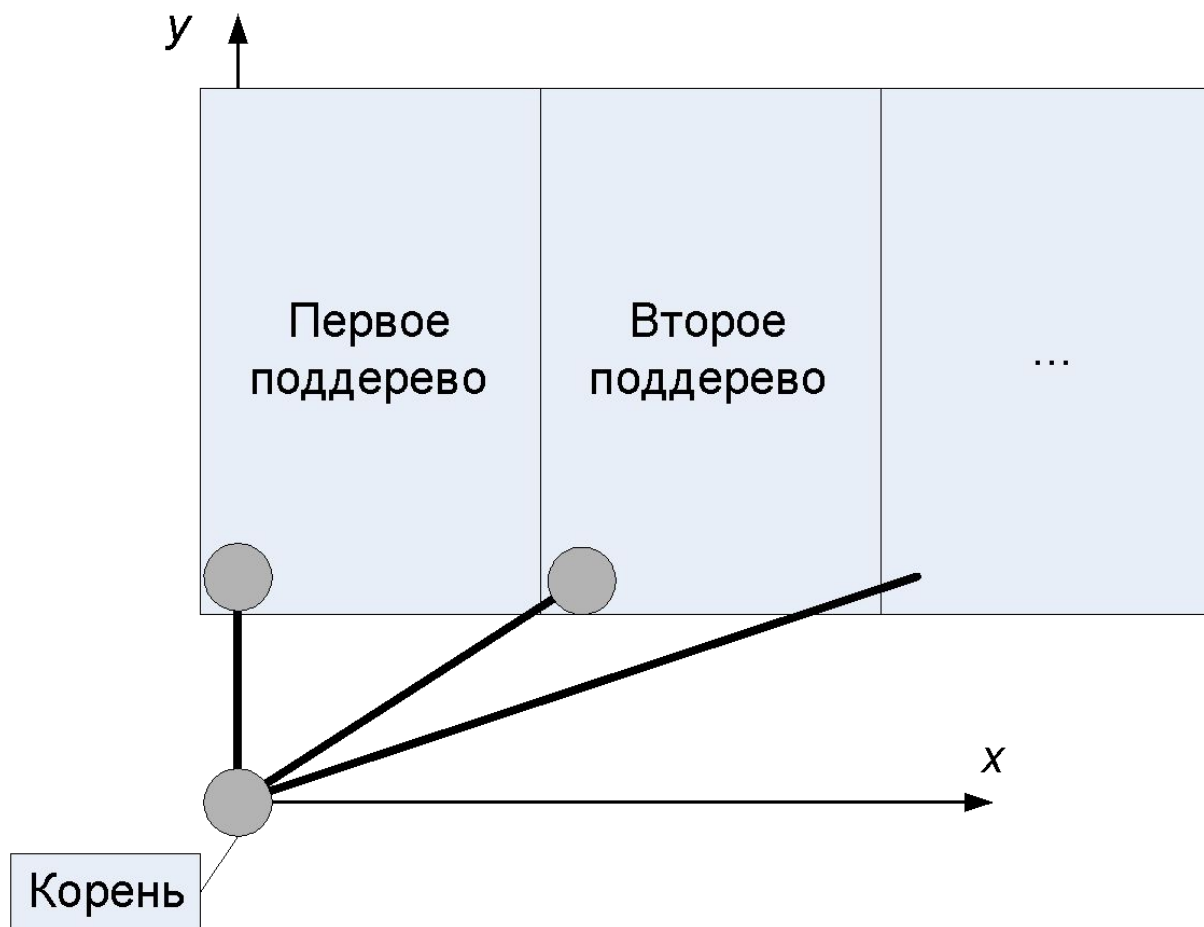
Разбор задачи – Федор Царев

Формальная постановка задачи

- Задан граф, в котором каждое ребро лежит не более чем на одном простом цикле – **реберный кактус**
- Необходимо расположить его вершины на плоскости в точках с целыми координатами, так чтобы ребра были представлены непересекающимися отрезками



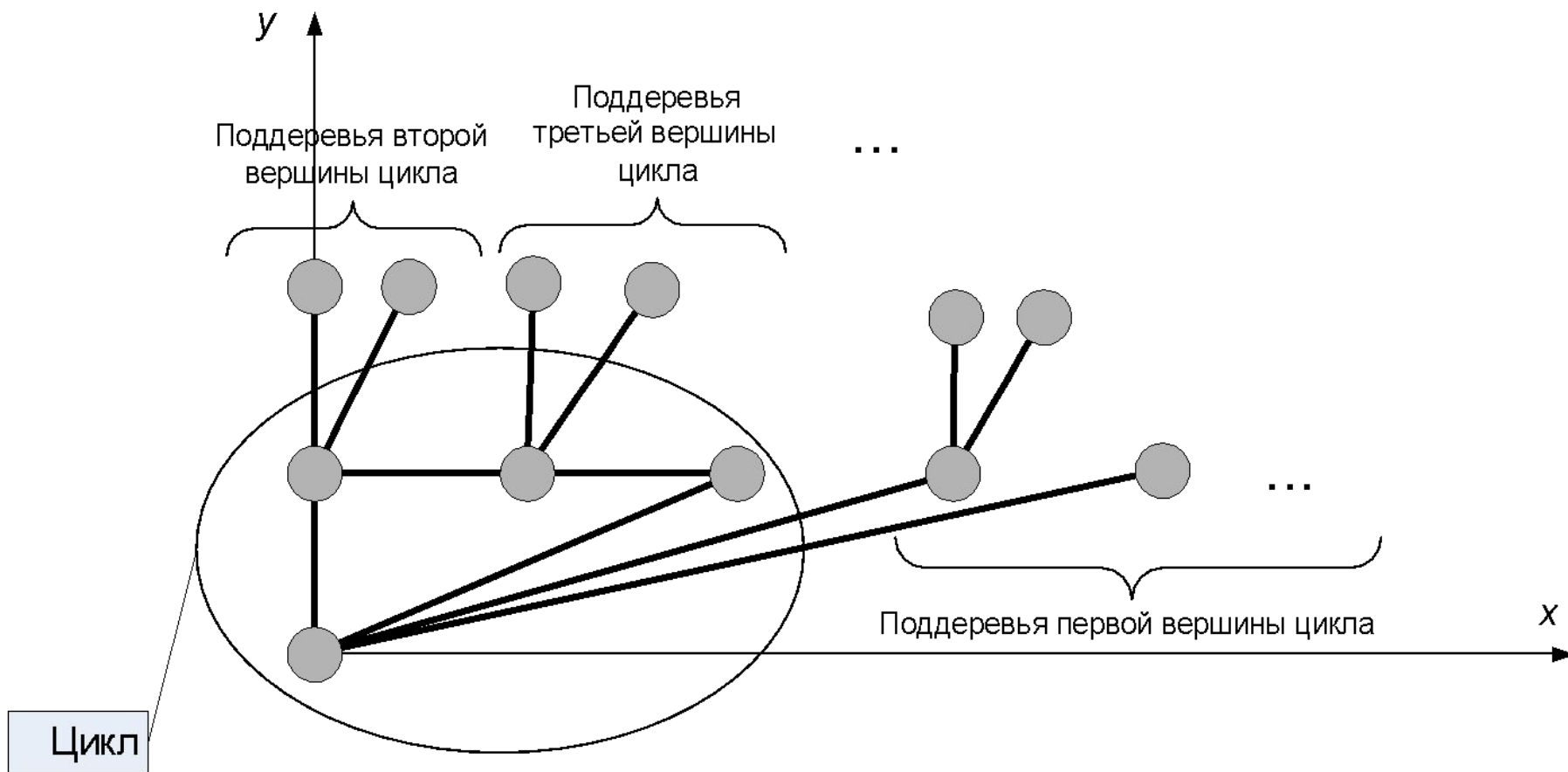
Нет циклов – 40 баллов



Как реализовать?

- Модификация алгоритма обхода в глубину
- Хранить максимальную X -координату уже поставленной вершины

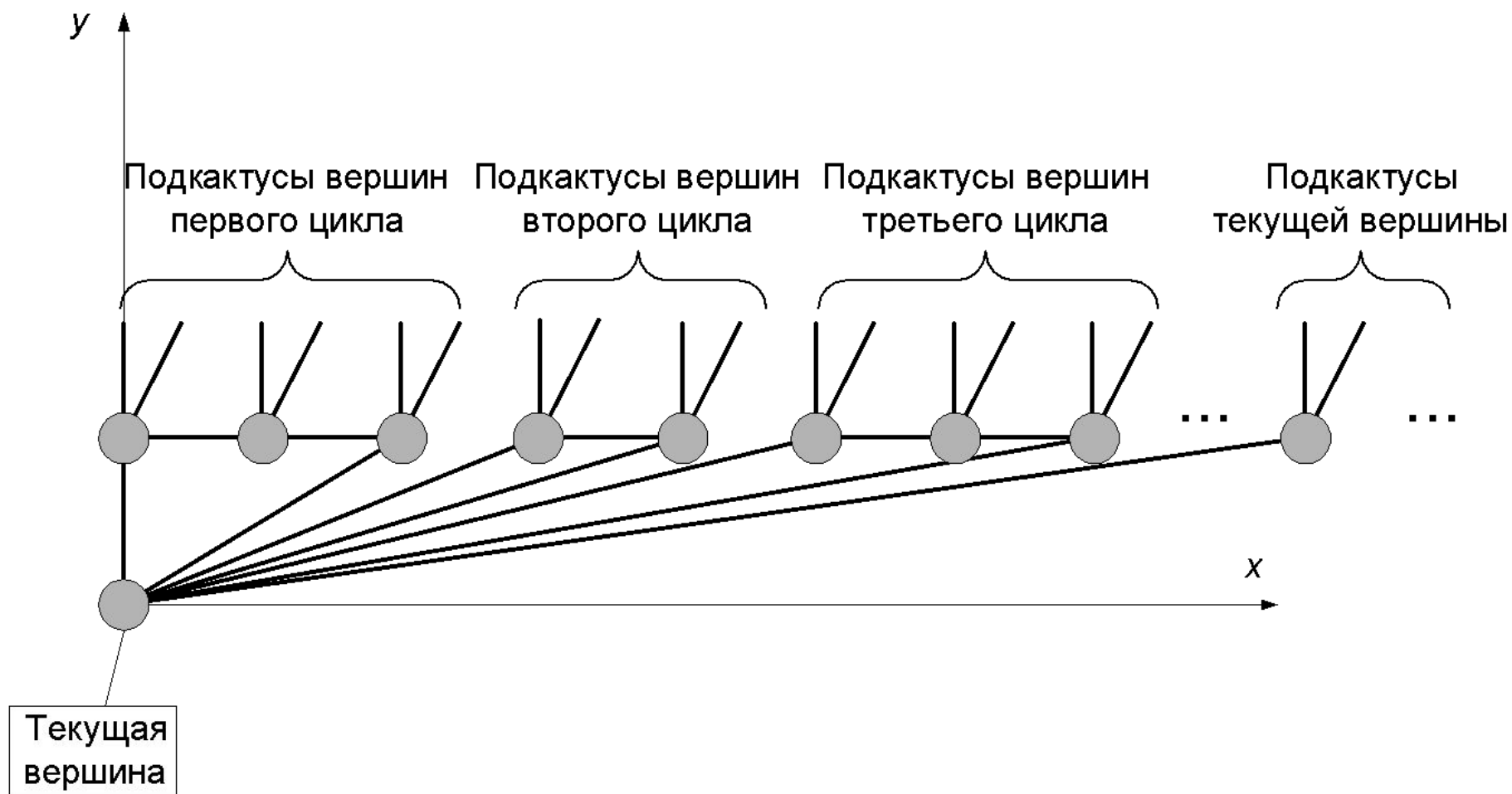
Один цикл – 60 баллов



Как реализовать?

- Найти цикл с помощью поиска в глубину
- Расположить одну из его вершин в начале координат
- Обработать остальные вершины цикла с их поддеревьями
- Расположить вершины поддеревьев первой вершины

Полное решение – 100 баллов



Как реализовать?

- Для каждой вершины найти список циклов, в которые она входит
- Это можно сделать с помощью обхода в глубину

80 баллов

- За прохождение всех тестов, в которых **каждая вершина лежит не более чем на одном простом цикле**
- Не нужно использовать список для хранения всех циклов, на которых лежит вершина

Система тестов

	$N \leq 10$	$N \leq 10^3$	$N \leq 10^5$	Всего баллов
Без циклов	10	20	10	40
Один цикл	4	8	8	20
Вершинный кактус	4	8	8	20
Реберный кактус	2	4	14	20
Всего баллов	20	40	40	100



Спасибо за внимание!
Вопросы?

Задача 2. НГУ-стройка



Автор задачи – Сергей Копелиович

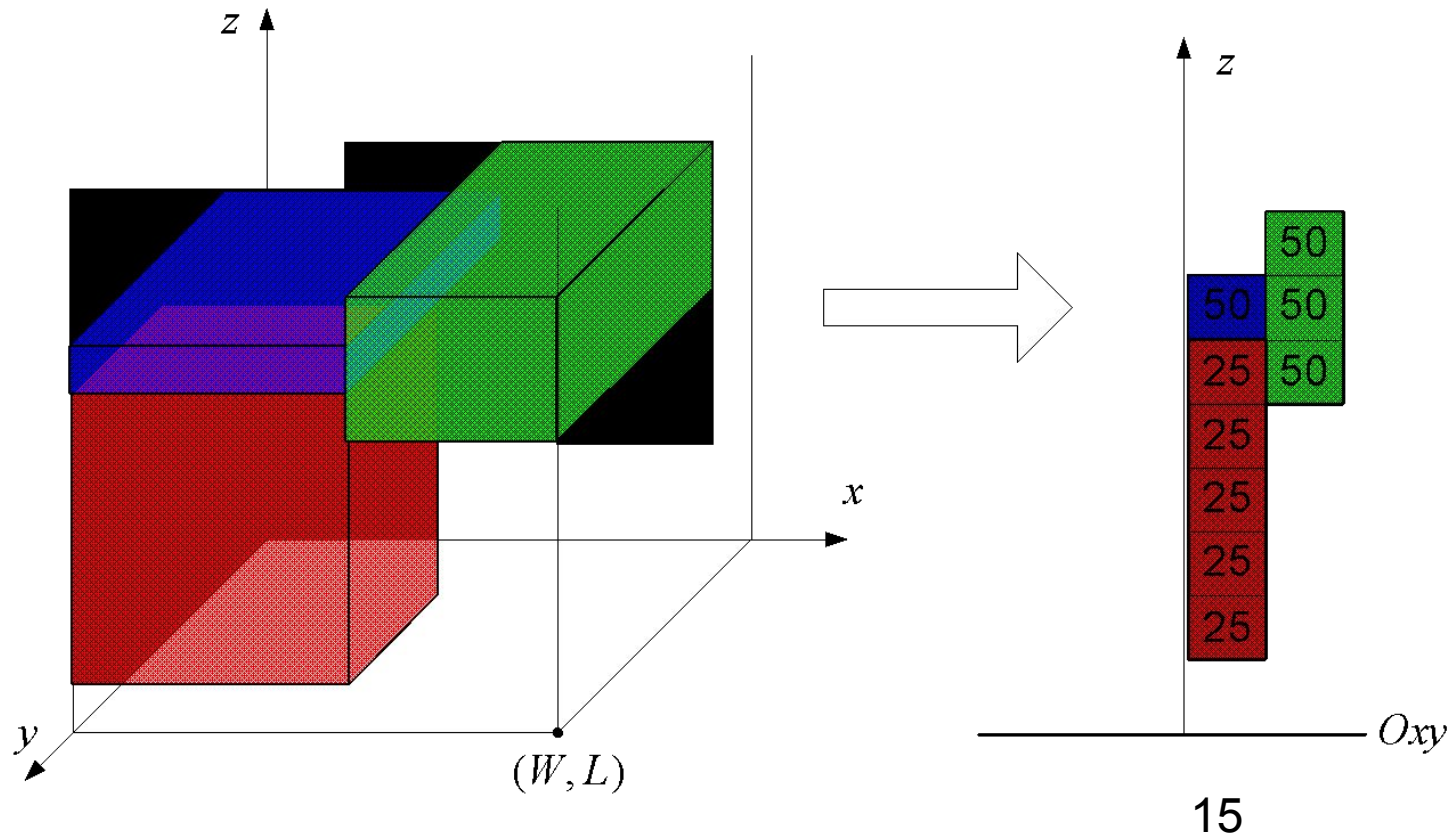
Разбор задачи – Сергей Назаров

Медленное решение

- «Нарисовать» блоки **в трехмерном массиве** логического типа и проверить каждый из горизонтальных слоев
- Время работы – $O(W \cdot H \cdot L)$, так как блоки **попарно не пересекаются**
- 40 баллов

Идея более быстрых решений – 1

- Спроецировать все блоки на ось Oz



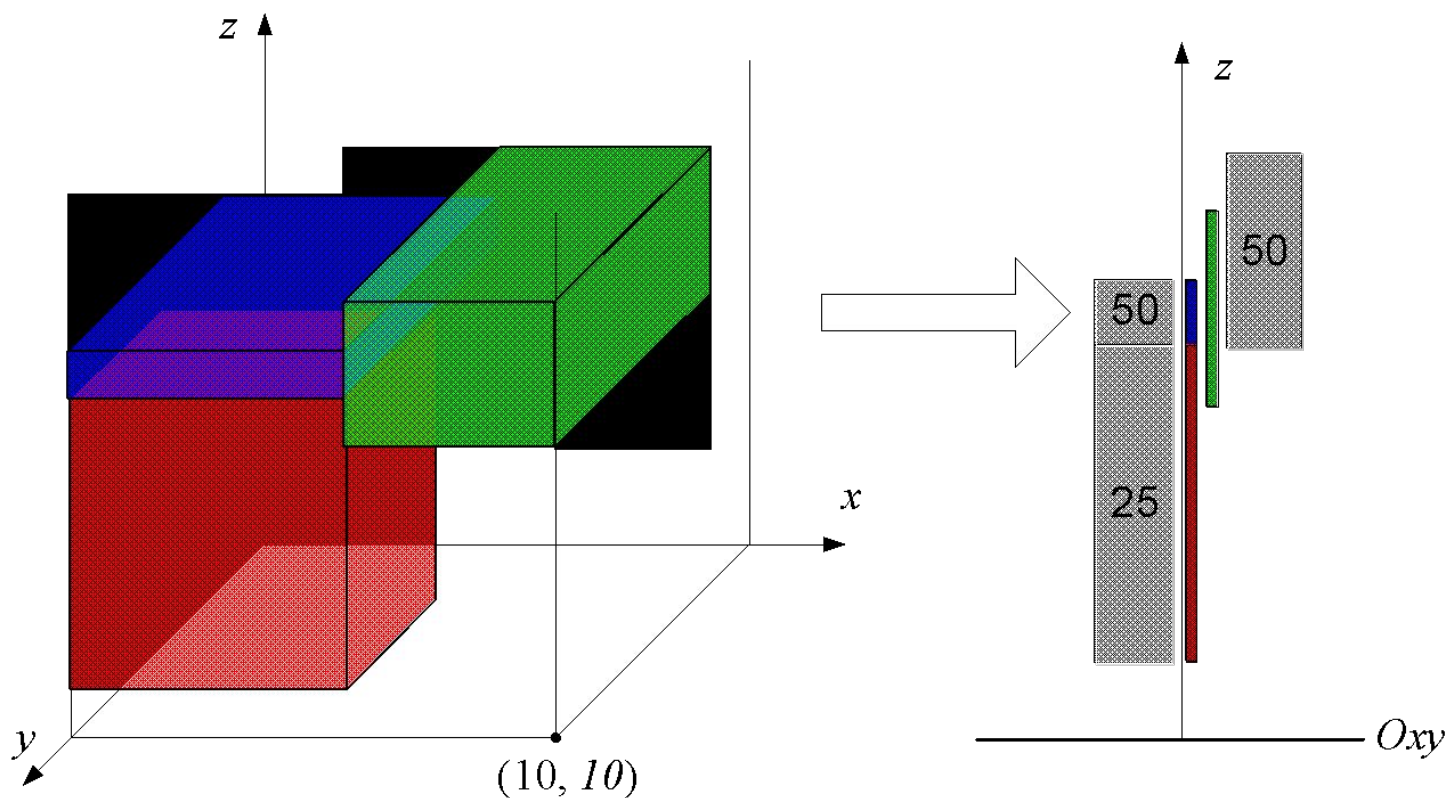
Идея более быстрых решений – 2

- Горизонтальный уровень $\leftrightarrow$ отрезок длиной 1 с целыми координатами концов
- Уровень заполнен $\leftrightarrow$ сумма чисел, соответствующих отрезкам, равна $W \cdot L$

К чему сводится задача?

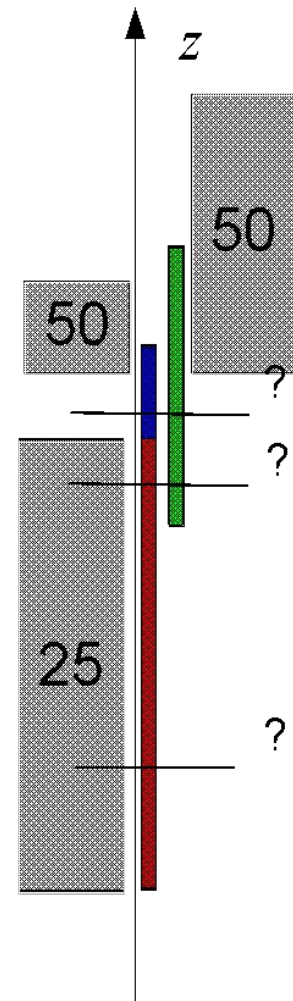
- Задан набор помеченных числами отрезков на прямой с концами в точках с целыми координатами
- Необходимо найти отрезок длиной 1, который покрыт **наименьшим** числом заданных отрезков, **сумма чисел на которых равна $W \cdot L$**

К чему сводится задача?



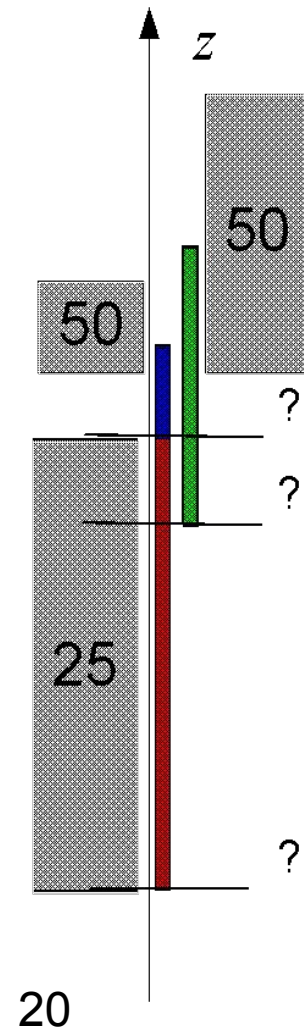
Более быстрое решение

- Попробовать в качестве ответа каждую целочисленную координату z
- Для каждой координаты z считать сумму площадей блоков и их количество
- Время работы – $O(N \cdot H)$
- 50 баллов



Еще более быстрое решение

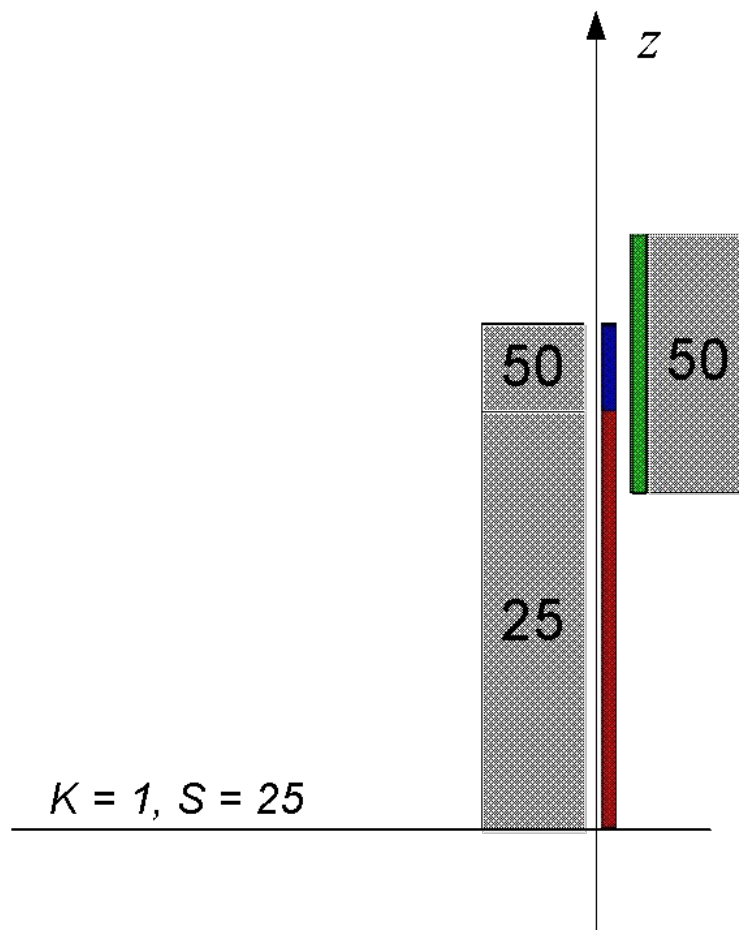
- Можно перебирать только координаты z , в которых начинается какой либо отрезок
- Для каждой координаты z считать сумму площадей блоков и их количество
- Время работы – $O(N^2)$
- 60 баллов



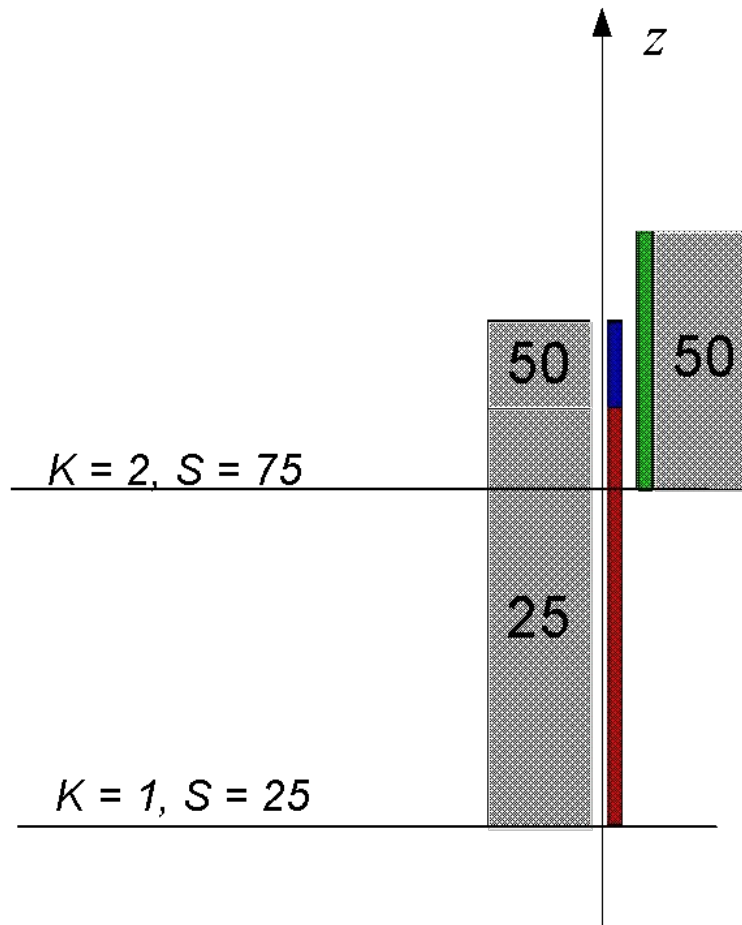
Техника «обработки событий»

- Движемся вдоль оси Oz снизу вверх
- События двух типов:
 - отрезок начался
 - отрезок закончился
- Поддерживаются значения двух величин:
 - **K – число отрезков**, которые уже начались, но еще не закончились
 - **S – сумма чисел**, соответствующих этим отрезкам

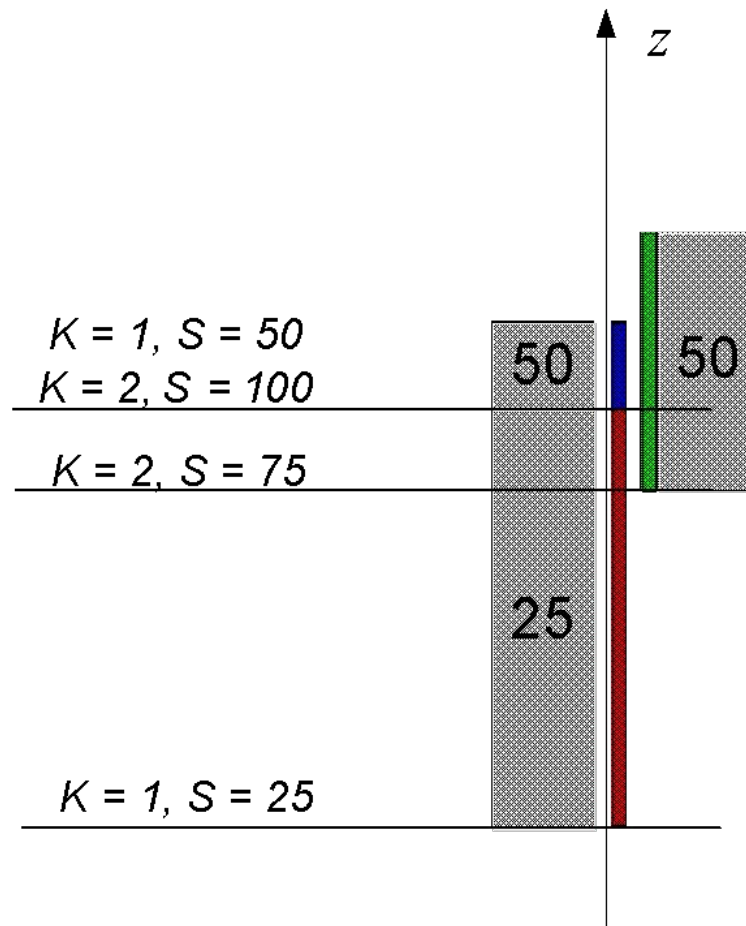
Техника «обработки событий»



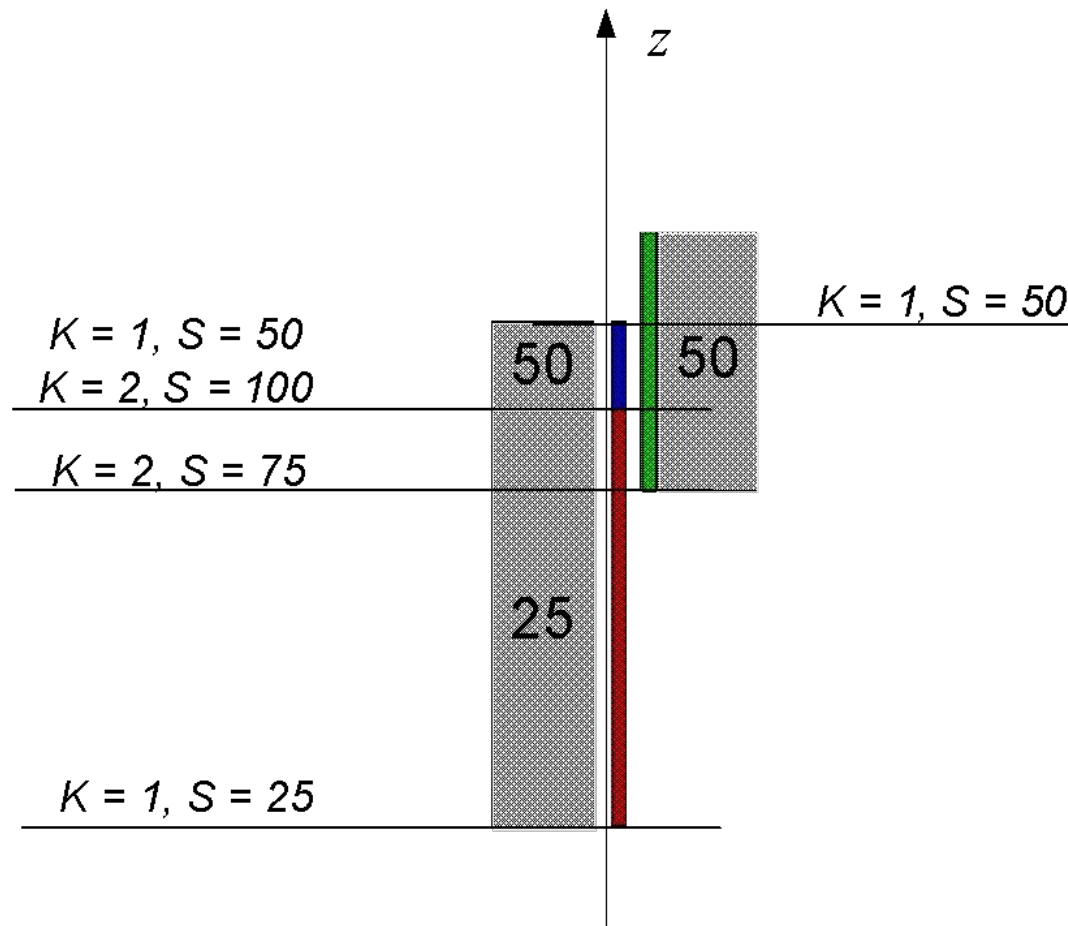
Техника «обработки событий»



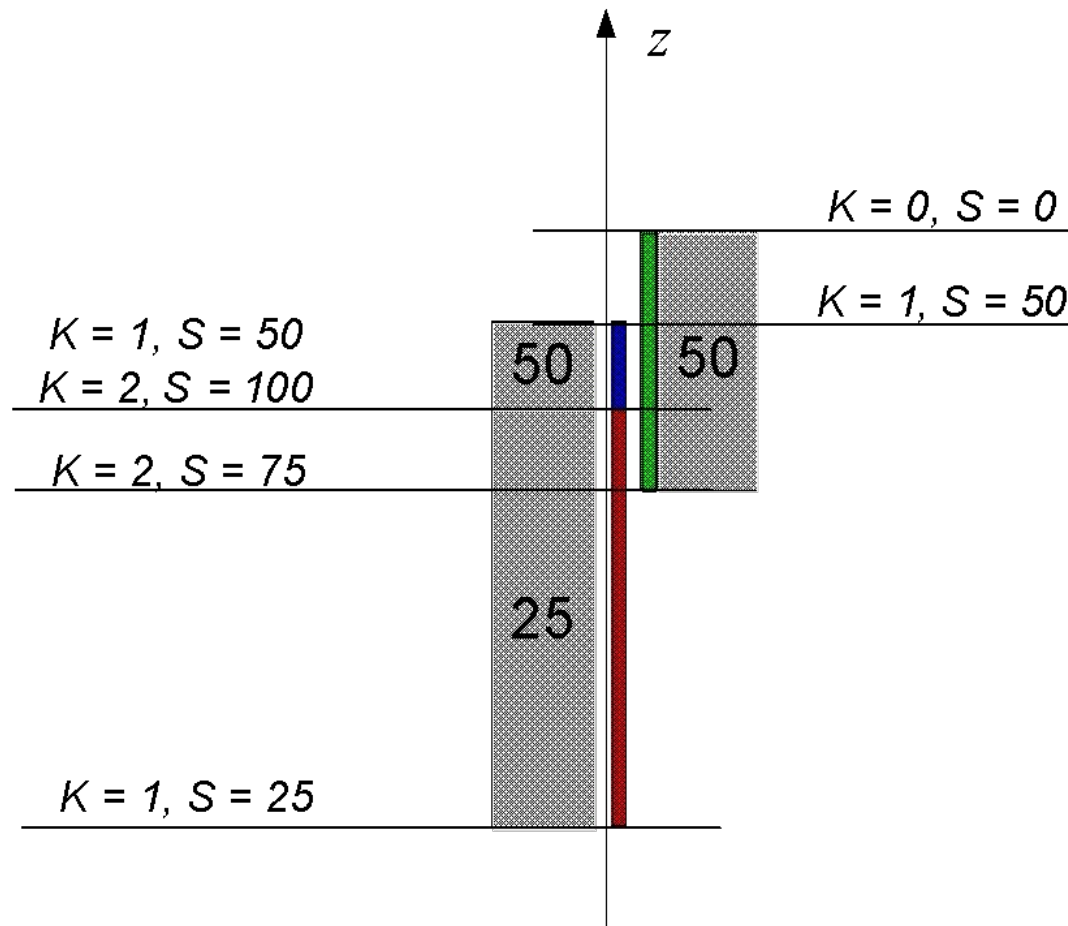
Техника «обработки событий»



Техника «обработки событий»



Техника «обработки событий»



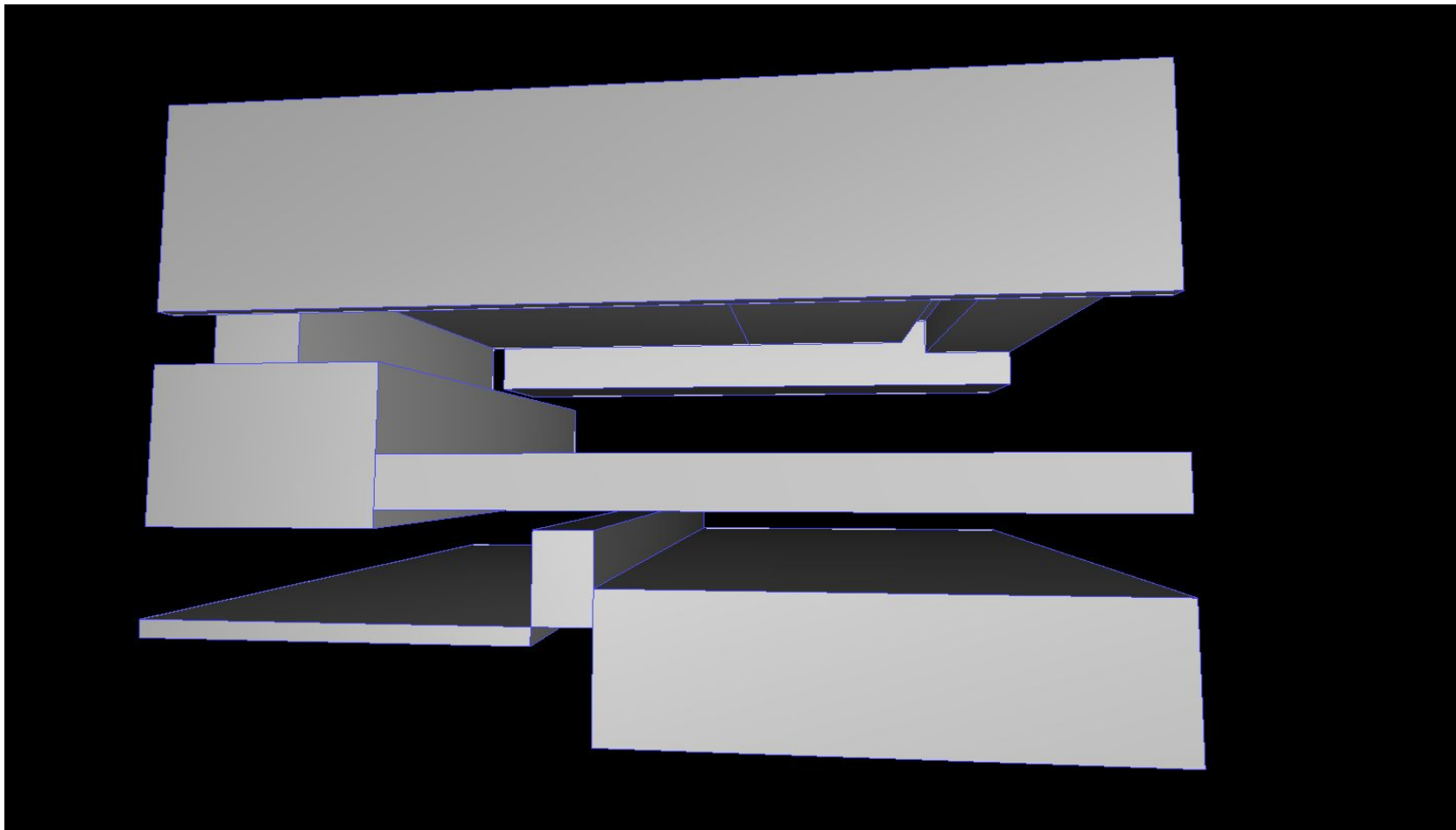
Сортировка событий

- Событие e характеризуется двумя числами: $e.z$ и $e.type$ (+1 – начало отрезка, -1 – конец отрезка)
- $e1 < e2$, если:
 - либо $e1.z < e2.z$
 - либо $(e1.z = e2.z)$ и $(e1.type < e2.type)$
- За $O(n^2)$ – 60 баллов
- За $O(n \cdot \log n)$ – 100 баллов

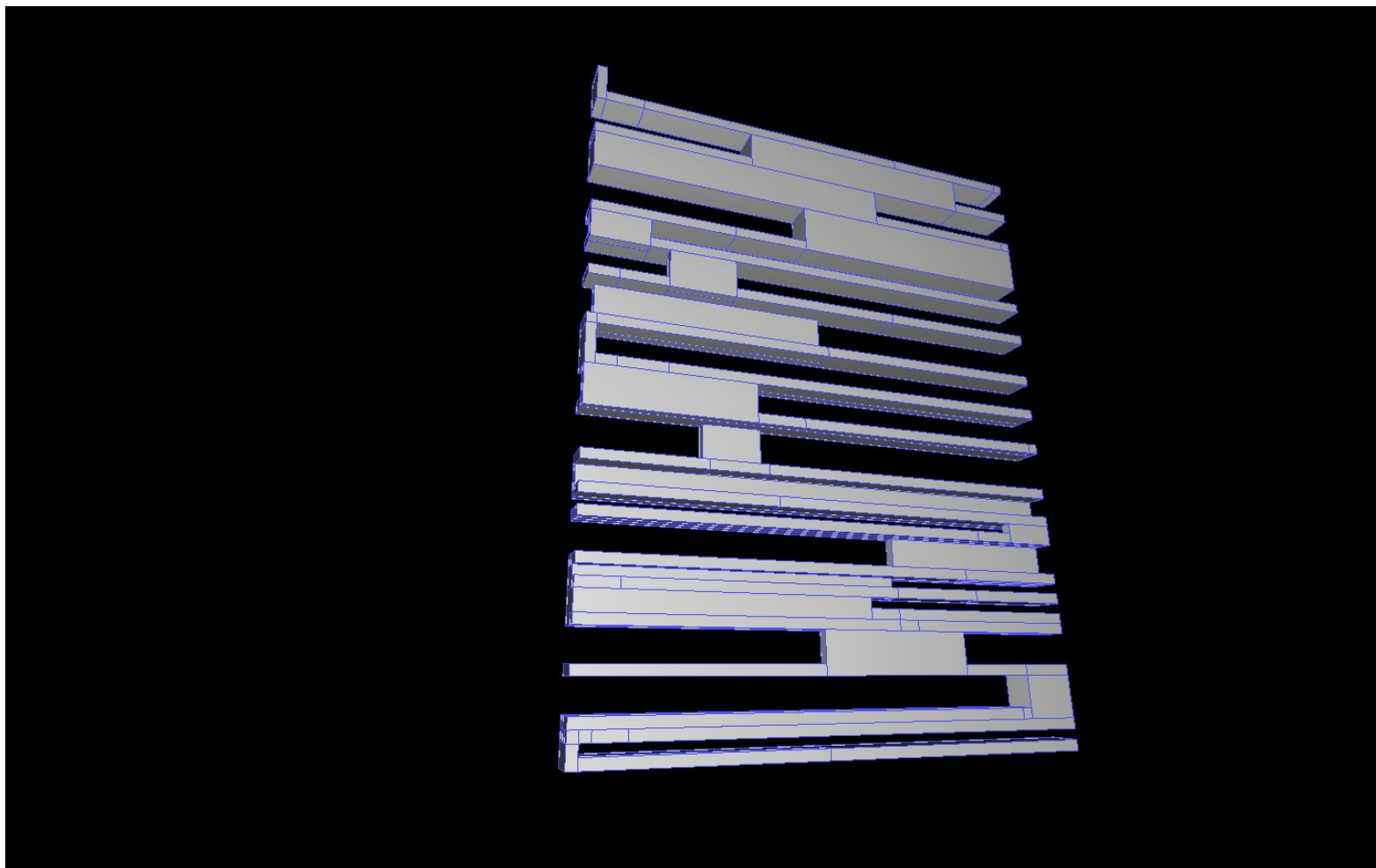
Еще одно решение

- Для каждой координаты хранить список событий, которые в ней происходят
- Время работы – $O(N + H)$
- 80 баллов

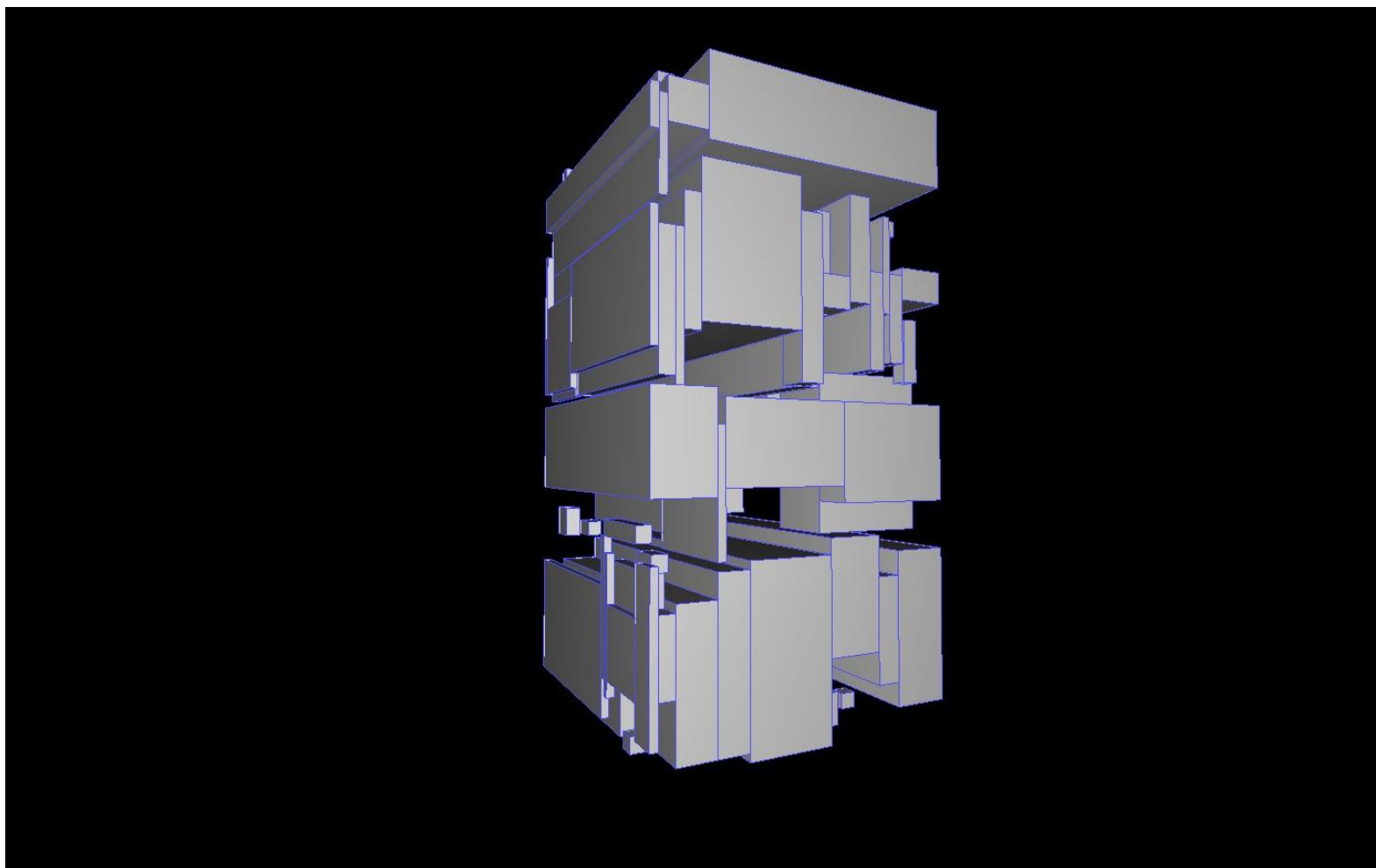
Пример теста – 1



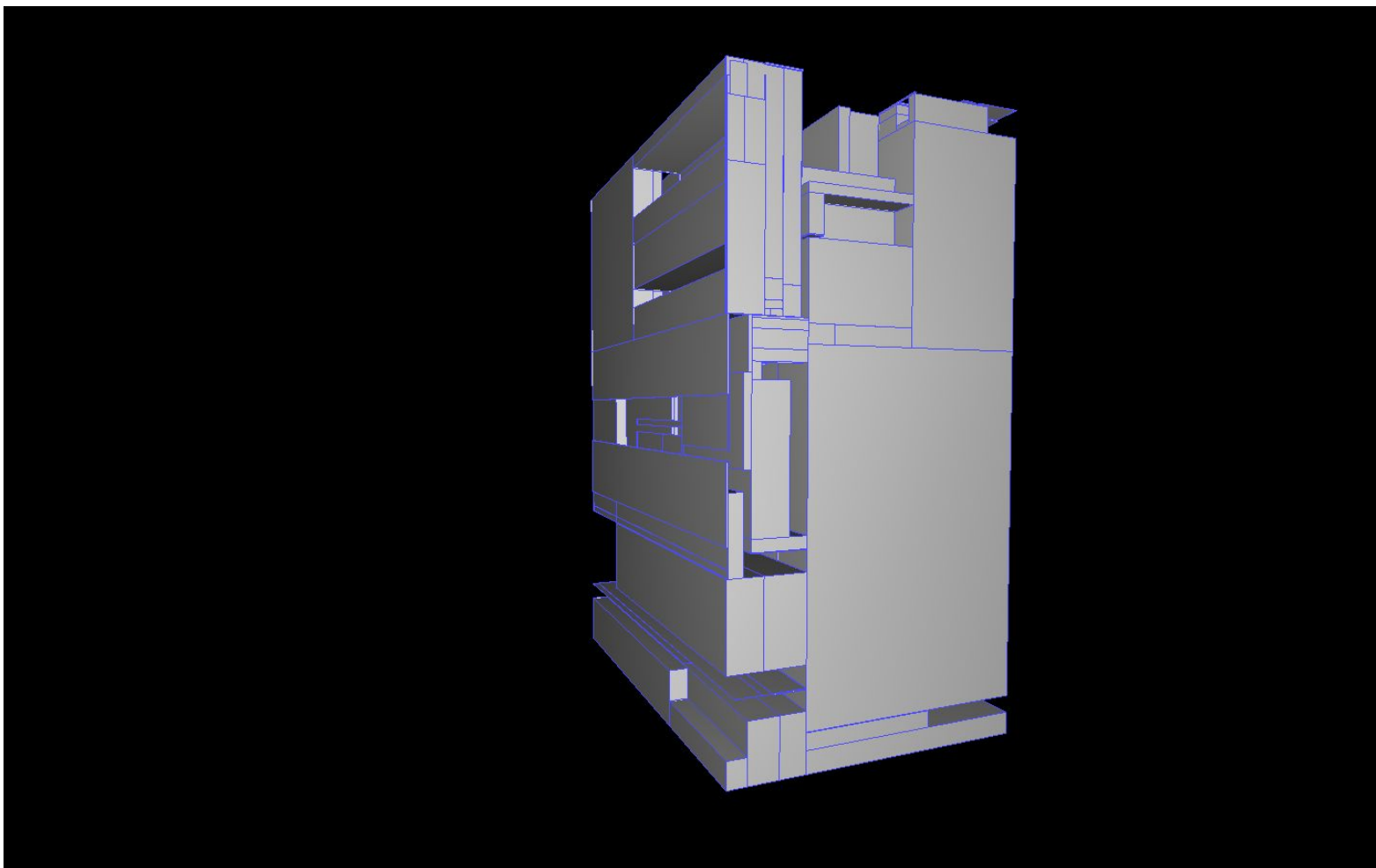
Пример теста – 2



Пример теста – 3



Пример теста – 4



Система тестов

- 1 – 20 тесты: $W \cdot H \cdot L \leq 10^6$
- 21 – 25 тесты: $N \cdot H \leq 10^6$
- 26 – 30 тесты: $N \leq 2500$
- 31 – 40 тесты: $N \leq 10^5$, $H \leq 2 \cdot 10^6$
- 41 – 50 тесты: $N \leq 10^5$, $H \leq 10^9$



Спасибо за внимание!
Вопросы?

Задача 3. Цифры и числа



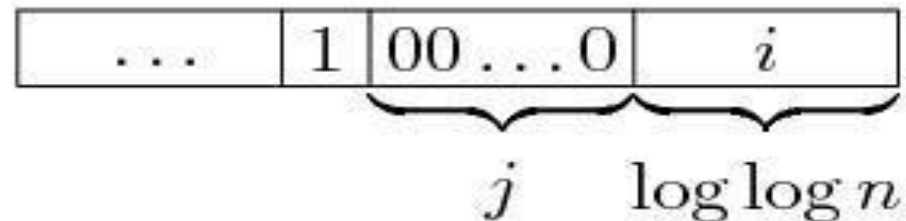
Автор задачи – В. А. Кузнецов

Разбор задачи – Илья Разенштейн

Простейший подход

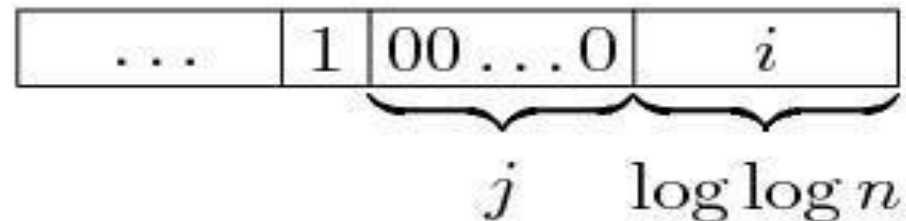
- Перебираем числа от 1 до n и для числа i вычеркиваем число $i + S(i)$
- Оптимизации:
 - битовый массив
 - ускоренный подсчет функции S
 - $S(k) = O(\log k)$
 - предварительные вычисления
- Диапазон баллов: 40 – 60

Полиномиальное решение – 1



- Поймем, какие биты могут измениться, если вычитать числа от 0 до $\log n$
- Разбиваем числа на классы (i, j, k) , где k – **количество** единиц в первой части числа.
- Возникают подзадачи: понять, является ли класс «некрасивым», и найти его мощность

Полиномиальное решение – 1



- Количество классов – $O(\log^3 n)$
- Проверка класса – $O(\log^2 n)$
- Время подсчета ответа для каждого класса – $O(\log n)$
- Итого: $O(\log^5 n)$. Кажется, что можно уменьшить время работы, но зачем? ☺

Полиномиальное решение – 2

- Техника «разделяй и властвуй»
- Подсчет количества «некрасивых» чисел на интервале $[0, 2^k)$
- Разбиваем наш интервал на две половины. Идея: структура «некрасивых» чисел на второй половине не сильно отличается от структуры на первой
- Пример: $k = 6$

01001010000001010010010100000010
10000101000000101001001010000001

Полиномиальное решение – 2

- На краях сказывается влияние первой половины. Чтобы это учесть, явно пересчитываем края. В середине сдвиг объясняется тем, что сумма цифр увеличивается на 1
- Обобщение данного метода для произвольных n – классическая конструкция в духе «получить номер по объекту»
- Итог: более простой алгоритм, который работает за $O(\log^3 n)$

```
01001010000001010010010100000010
10000101000000101001001010000001
```


Система тестов

- 50 тестов – за прохождение каждого из них по два балла
- 1 – 20 тесты: $1 \leq n \leq 10^6$
- 21 – 30 тесты: $10^8 \leq n \leq 10^{11}$
- 31 – 40 тесты: $10^{12} \leq n \leq 10^{14}$
- 41 – 50 тесты: $10^{14} \leq n \leq 10^{18}$



Спасибо за внимание!
Вопросы?

Задача 4. Легкоатлетический манеж НГУ



Автор задачи – В. А. Кузнецов

Разбор задачи – Андрей Станкевич

Наблюдение – 1

- Если $n(n+1)/(2m)$ не является целым числом, то решения не существует
- Если $n(n+1)/(2m) < n$, то самая большая полоса тартана не помещается на дорожку, поэтому решения не существует.
- Оказывается, эти два условия – необходимый и достаточный критерий отсутствия решения

Жадный алгоритм

- Пытаемся покрыть дорожку, покрывая остаток самой большой из оставшихся полос тартана, которая помещается
- Тест $n = 23$, $m = 6$
- $23 * 24 / (2 * 6) = 46$
 - $23+22+1$
 - $21+20+5$
 - $19+18+9$
 - $17+16+13$
 - $15+14+12+4+?$

Преобразование жадного алгоритма в перебор

- Тест $n = 23, m = 6$
- $23 * 24 / (2 * 6) = 46$
 - $23+22+1$
 - $21+20+5$
 - $19+18+9$
 - $17+16+13$
 - $15+14+12+2+3$
 - $11+10+8+7$
- Если не получилось, попробуем другой вариант

Перебор

- Заполняем дорожки по очереди
- На каждом шаге пытаемся положить на текущую дорожку еще одну полосу, начиная с самой большой полосы, которая туда помещается и еще не использована
- Если не получилось – выполняется возврат

Наблюдение 2

- Если задача имеет решение для чисел m и n , то она имеет решение и для чисел m и $(n+2m)$
 - Удлиним каждую дорожку на $2n+2m+1$, используя для покрытия новой части пары $(n+1, n+2m)$, $(n+2, n+2m-1), \dots, (n+m, n+m+1)$
- Таким образом, достаточно найти решение для минимального n_0 , имеющего такой же остаток по модулю $2m$, что и число n

Решение

- Перебираем в порядке возрастания такие числа n_0 , что $n_0 \geq 2m-1$ и n_0 имеет тот же остаток по модулю $2m$, что и n
- Для каждой пары (m, n_0) пытаемся распределить полосы тартана по дорожкам с помощью перебора
- Как только нашли решение – останавливаемся
- Дополняем решение для (m, n_0) до решения для (m, n)

Полное решение?

- Набирает 90 баллов
- Не проходит тесты:
 - $m = 957, n = 3740$
 - $m = 979, n = 3827$
 - получающиеся из этих прибавлением $2m$ к n

Что делать?

- Запустить решение на всех тестах – их 6832
- Заметить, что решение работает долго (порядка 10 секунд) только на этих тестах
- Предподсчитать ответы на эти тесты

Что можно еще сделать?

- Добавить в перебор элементы случайного поиска
- Если решение на задачу есть, то решений достаточно много
- Можно найти ветку перебора, в которой решение будет найдено быстро

Если вы не сделали наблюдение 2

- Если задача имеет решение для чисел m и n , то она имеет решение и для чисел m и $(n+2m)$
- Ничего страшного – такое решение проходит тесты ($m = 957, n = 5654$) и ($m = 979, n = 5785$), которые не проходит решение с использованием этого наблюдения 😊
- 90 баллов

«Жадные» решения

- Различные варианты жадного алгоритма – около 30 баллов
 - Заполнять дорожку, где остался максимум
 - Добавлять в текущую дорожку максимальную возможную полосу тартана
 - ...
- Добавлять одну полосу в ту дорожку, которую она либо полностью заполнит, либо (если таких нет) в самую свободную – 88 баллов (если две – то 100 баллов 😊)

Математическое решение за $O(n+m)$

- Рассмотрим $S=n(n+1)/(2m)$, $S < 2n$
- Рассмотрим пары $(n, S-n)$, $(n-1, S-n+1), \dots$
 - Если S нечетно, то последняя пара $((S-1)/2, (S+1)/2)$
 - Остаются числа $\{1, 2, \dots, S-n-1\}$
 - Их можно разбить по индукции
 - Если S четно, то последняя пара $(S/2-1, S/2+1)$
 - Остаются числа $\{1, 2, \dots, S-n-1, S/2\}$
 - Разобьем по индукции $\{1, 2, \dots, S-n-1\}$ на части размера $S/2$
 - Их получится нечетное число, вместе с $S/2$ можно составить оставшиеся части

Система тестов

- Содержит тесты против:
 - жадных решений (35 тестов)
 - неоптимальных реализаций перебора (10 тестов)
 - против «правильных» переборов (по 5 тестов)



Спасибо за внимание!
Вопросы?

Задача 5. Граффити на заборе

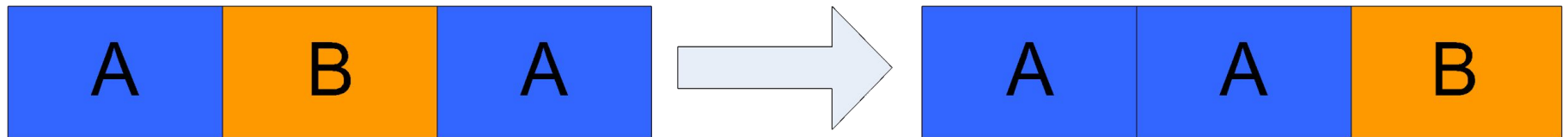


Автор задачи – Денис Денисов

Разбор задачи – Денис Денисов

Идеи решения

- Каждый граффити-художник разрисовывает сплошной отрезок плит забора
- Если исходно художник i находится левее художника j , то и плиты, которые он разрисовывает находятся левее



Двоичный поиск по ответу

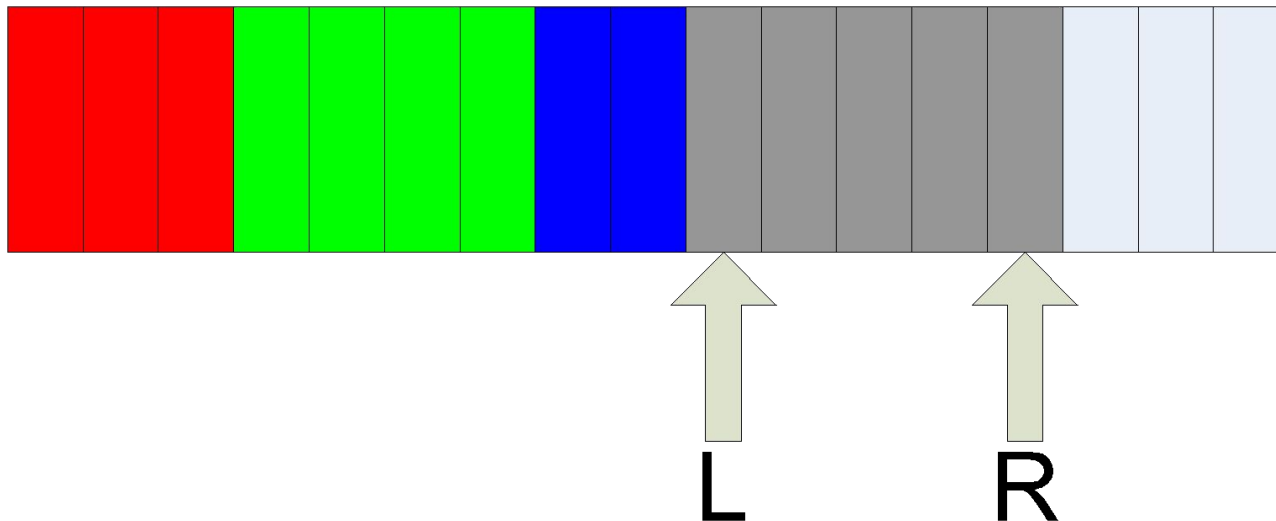
- Если художники могут разрисовать забор за T минут, то могут и за $T+1$ минуту
- Решим более простую задачу – проверим, могут ли художники разрисовать все плиты за не более, чем T минут
- После этого можно организовать двоичный поиск по T

Решение более простой задачи – 1

- «Жадный» алгоритм
- Каждый художник разрисовывает столько плит, сколько он успевает за время T

Решение более простой задачи – 2

- Правую границу можно найти линейным поиском, добавляя плиты по одной пока художник успевает их разрисовать
- $t_i = (\min(|p_i - L|, |p_i - R|) + (R - L)) \cdot a + (R - L + 1) \cdot b$



Оценка времени работы – 1

- Предварительная сортировка художников по неубыванию начальной позиции – $O(m \cdot \log m)$ (или $O(n+m)$ при использовании сортировки «подсчетом»)
- Верхняя оценка на время покраски – $(2a+b)n$

Оценка времени работы – 2

- Время работы «жадного» алгоритма – $O(n+m)$ – так как на каждом шаге либо разрисовывается еще одна плита, либо происходит переход к следующему художнику
- Итого: $O(m \cdot \log m + (n+m) \cdot \log((2a+b)n))$
- Такое решение получает 100 баллов

Другие варианты

- Искать новую правую границу на каждом шаге двоичным поиском – 100 баллов
- Искать правую границу, решая линейные неравенства – 100 баллов
- Время работы этих решений пропорционально $\log n$ – они работают и при гораздо больших значениях n

Частичное решение – 1

- Динамическое программирование ($O(n^2m)$)
- $F[i][j]$ – минимальное время, которое требуется первым i художникам на разрисовывание первых j плит своими рисунками
- 40 баллов

Частичное решение – 2

- Обработка случая $m \leq 2$
- Перебрать точку разделения отрезков и по формуле найти время, требуемое для разрисовывания
- 40 баллов

Что можно забыть?

- Сортировку по неубыванию – получите не больше 30 баллов
- Быстрые алгоритмы сортировки – получите не больше 80 баллов
- 64-битный тип данных – получите не больше 80 баллов:
 - `int64` в Delphi
 - `long long` в C++

Система тестов

	$M \leq 2$	$M \leq 100$	$M \leq 10^5$	Всего
$N \leq 100$	30	10	0	40
$N \leq 10^5$	10	4	46	60
Всего	40	14	46	100



Спасибо за внимание!
Вопросы?

Задача 6. Стековый калькулятор



Автор задачи – Денис Денисов

Разбор задачи – Сергей Копелиович, Виталий Вальтман

Идея решения

- Динамическое программирование
- Поиск кратчайшего пути в графе

40 баллов

- Вершина графа – набор чисел в стеке
- Ребра графа соответствуют операциям из условия задачи
- Поиск в ширину
- При реализации граф полностью строить не нужно, по ходу поиска в ширину будут появляться новые вершины

50 баллов

- Состояние – пара (n, k)
- Переходы:
 - $f(n, k) = \min(f(a, k) + f(b, k-1) + 1)$, где a, b – такие числа, что либо $a+b=n$, либо $a-b=n$, либо $a*b=n$
- Числа, большие чем $2N$, использовать не надо
- Динамическое программирование?

Нет!

- Есть циклы:

- $f(70, k) \leftarrow f(75, k) + f(5, k-1) + 1$
- $f(75, k) \leftarrow f(70, k) + f(5, k - 1) + 1$

- Итерационный алгоритм:

- изначально $f(1, k) = 1$ для всех $k > 0$, остальные $f(n, k) = 10^9$
- улучшаем по формуле $f(n, k) = \min(f(a, k) + f(b, k-1) + 1)$

- Форд-Беллман $O(TE)=O(KN^2\log N)$

- Дейкстра $O(V^2)=O((NK)^2)$

Улучшение до $O(N \log^2 N)$

- Всегда хватит $K \leq 5$
- Число переходов типа «умножение» – $O(N \log N)$
- Число переходов типа «сложение» и «вычитание» – $O(N^2)$
- $f(N, 5) \approx 4 \cdot \log_2 N \rightarrow$ при сложении и вычитании одно из двух чисел должно быть «маленьким». «Маленькие» = $\{-3, -2, -1, 0, 1, 2, 3\}$
- $O(N \log N)$ переходов и $O(N)$ состояний
- У нас уже 80 баллов!!!

Полное решение

- $f(a,k)=2a-1$, для $a \leq 3$
- Сложение и вычитание:
 - $f(n,k) \leftarrow f(n+a,k)+2|a|$, для $|a| \leq 3$
- Объединяем с умножением:
 - $f(n,k) \leftarrow f(x,k) + f(y,k-1) + 2|a| + 1$, где $x*y=n+a$ и $|a| \leq 3$
- $n > x$, $n > y$ при $n > 5$ – поэтому граф ацикличесен
- Динамическое программирование?



Да!!!

- Есть решение за $O(N \log N)$
- Конечное состояние достижимо из $O(\sqrt{N})$ состояний
- «Ленивая динамика» с конца
- 100 баллов

Система тестов

	$N \leq 100$	$N \leq 1000$	$N \leq 10^5$	$N \leq 10^6$	$N \leq 10^9$	Всего
$K=2$	4				2	6
$K=3$	16	4	6	6	2	34
$K=4$	8		2	6	6	22
$K=5$	2				6	8
$K=6$	2					2
$K=7..10$	8	2				10
$K=11..100$		4	2	8	4	18
Всего	40	10	10	20	20	100



Спасибо за внимание!
Вопросы?