

Всероссийская олимпиада школьников по  
информатике  
Региональный этап, II тур  
Разбор задач

Андрианов И. А.  
Стрекаловский О. А.

Вологодский государственный университет  
Факультет прикладной математики,  
компьютерных технологий и физики

Вологда  
2015 г.

- Согласно условию  $0 \leq D_a, D_b \leq N/2$
- Обозначим станцию Андрея  $A$ , а станцию Бориса  $B$
- Если для двух выбранных станций  $A$  и  $B$  существует хотя бы одна пара  $[D_a, D_b]$  такая, что среди станций  $\{A + D_a, A - D_a, B + D_b, B - D_b\}$  есть две или более совпадающих, то станции  $A$  и  $B$  не могут быть выбраны
- Если станция не определяется однозначно по  $[D_a, D_b] \Rightarrow$  существует хотя бы две станции, до каждой из которых можно доехать из  $A$  за  $D_a$  перегонов и из  $B$  за  $D_b$  перегонов
- Очевидно, что  $A$  и  $B$  не должны располагаться за равное число перегонов по часовой стрелке и против часовой стрелки

Рассмотрим случай, когда количество станций  $N$  нечетно.

- Пусть Андрей живёт на любой из  $N$  станций
- Зафиксируем станцию, на которой живёт Андрей. Это можно сделать  $N$  способами
- Борис не может жить только на станции  $A$ , т.к. только при 0 перегонов между  $A$  и  $B$  может быть одинаковое расстояние в обоих направлениях  $\Rightarrow$  Борис может жить на любой из оставшихся  $N - 1$  станций
- Итого  $N * (N - 1)$  пар станций таких, что любая станция определяется уникальной парой расстояний  $[D_a, D_b]$  при фиксированных номерах  $A$  и  $B$

Рассмотрим случай, когда количество станций  $N$  чётно.

- Пусть Андрей живёт на любой из  $N$  станций
- Зафиксируем станцию, на которой живёт Андрей. Это можно сделать  $N$  способами
- Борис не может жить только на станциях  $A$  и  $A + N/2$ , т.к. только при таком размещении между  $A$  и  $B$  может быть одинаковое расстояние в обоих направлениях  $\Rightarrow$  Борис может жить на любой из оставшихся  $N - 2$  станций
- Итого  $N * (N - 2)$  пар станций таких, что любая станция определяется уникальной парой расстояний  $[D_a, D_b]$  при фиксированных номерах  $A$  и  $B$

### Типичные ошибки

- Неэффективное решение.  
Поиск всех подходящих пар станций через их перебор

Вопросы по задаче?

# «Вырубка леса»

## Основная идея

Двоичный поиск по минимальному количеству дней, которое хватит для вырубки всех деревьев при одновременной работе обоих лесорубов.

- Минимально может потребоваться один день  $\Rightarrow$   
левая граница поиска 0
- Максимально может потребоваться  $2 * X / (A + B)$  дней,  
если бы лесорубы работали бы с минимальным рабочим  
интервалом в 1 день  $\Rightarrow$   
правая граница поиска  $2 * X / (A + B) + 1$



## «Вырубка леса»

Вычисление значения функции в фиксированной точке

«Сколько деревьев срубят первый и второй лесоруб за фиксированное количество дней»

- За  $N$  дней первый лесоруб, срубая  $A$  деревьев в день, и, отдыхая каждый  $B$ -й день, срубает  $(N - \lfloor N/B \rfloor) * A$  деревьев
- За  $N$  дней второй лесоруб, срубая  $C$  деревьев в день, и, отдыхая каждый  $D$ -й день, срубает  $(N - \lfloor N/D \rfloor) * C$  деревьев

Складываем показатели обоих лесорубов

## «Вырубка леса»

Итоговый код двоичного поиска на Java

```
long left = 0;
long right = 2 * X / (A + C) + 1;
while (right - left > 1) {
    long days = (left + right) >>> 1;
    long removedByFirst = removedTrees(A, B, days);
    long removedBySecond = removedTrees(C, D, days);
    if (removedByFirst + removedBySecond >= X) {
        right = days;
    } else {
        left = days;
    }
}
return right;
```

«А. С. Станкевич. Лекция 2: Сортировка и поиск».<sup>1</sup>

<sup>1</sup>[http://www.youtube.com/watch?v=qkLLcdgJj\\_o](http://www.youtube.com/watch?v=qkLLcdgJj_o)

## Типичные ошибки

- Неэффективное решение.  
Линейный поиск дня окончания работ
- Использование 32-х разрядного типа данных
- Переполнение 64-х разрядного типа данных при подсчёте числа вырубаемых деревьев при фиксированном количестве дней
- В случае совпадения дней отдыха при полной вырубке деревьев необходимо брать предыдущий день
- Ошибки в реализации двоичного поиска (дают TLE/WA):
  - Неправильная оценка границ поиска
  - Неправильное сужение границ поиска
  - Неправильный выбор ответа, когда поиск останавливается

Вопросы по задаче?

Динамическое программирование по профилю.

$count[i][j][k]$  - количество способов заполнить первые  $i - 1$  столбцов плитками при профиле  $i$ -ого столбца  $[j][k]$ .

В профиле  $[j][k]$

$j = 1$ , если в верхней строке  $i$ -ого столбца ячейка заполнена,

$k = 1$ , если в нижней строке  $i$ -ого столбца ячейка заполнена.

- 1 Фиксированной плиткой назовём плитку, которую уже положили строители
- 2 Рассмотрим случай, когда хотим уложить в  $i$ -ый столбец вертикальную плитку размера 2
- 3 Мы можем это сделать, если у нас нет фиксированных плиток в  $i$ -ом столбце
- 4 Если мы это сделать можем, то  $count[i - 1][1][1]$  способом

Рассмотрим какие варианты положить плитки вверх и вниз ещё существуют.

- 1 В верхнюю строку можно ничего не класть, либо положить плитку длиной 1, либо плитку длиной 2
- 2 Аналогично, в нижнюю строку можно ничего не класть, либо положить плитку длиной 1, либо плитку длиной 2
- 3 Некоторые из этих вариантов невозможны из-за фиксированных плиток
- 4 Если вариант возможен, то смотрим, какой профиль получается в  $i$ -ом столбце и из какого в  $(i - 1)$ -ом столбце

## «Укладка плитки»

Код обчета  $i$ -ого столбца на C++

```
if (!singles[i][0] && !singles[i][1])
    count[i][1][1] = count[i - 1][1][1];
for (int topTileLen=0; topTileLen<=2; topTileLen++) {
    if (topTileLen!=1 && singles[i][0])
        continue;
    for (int bottomTileLen=0; bottomTileLen<= 2;
         bottomTileLen++) {
        if (bottomTileLen!=1 && singles[i][1])
            continue;
        int j1 = (topTileLen == 0) ? 0 : 1;
        int k1 = (bottomTileLen == 0) ? 0 : 1;
        int j2 = (topTileLen == 2) ? 0 : 1;
        int k2 = (bottomTileLen == 2) ? 0 : 1;
        count[i][j1][k1] += count[i - 1][j2][k2];
        count[i][j1][k1] %= 1000000007;
    }
}
```



### Типичные ошибки

- Неэффективное решение.  
Перебор вариантов укладки плиток
- Учёт не всех случаев расположений плиток
- Использование модульной арифметики только при выводе ответа, но не в промежуточных расчетах

Вопросы по задаче?

Спасибо за внимание!