Northern Eurasia Contests
Northern Subregional Contest 2018

Геннадий Короткевич

27 октября 2018 года

Problem A. Accumulator Battery

- Идея — Борис Минаев
- Разработка — Григорий Шовкопляс

Постановка задачи

- Батарея телефона разряжается с постоянной скоростью
- При 20% заряда и менее — вдвое медленнее
- За t минут батарея села со 100% до p%
- Через сколько минут батарея полностью сядет?

Решение

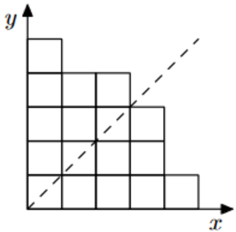
- Представим, что каждый процент ниже 20% равен двум, а всего в батарее 120%
- Теперь батарея всегда разряжается с постоянной скоростью и садится на уровне в -20%
- Если $p < 20$, переведём процент в новую шкалу по формуле $p := 2p - 20$, иначе оставим как есть
- Теперь ответ равен $t \cdot \frac{20+p}{100-p}$
- Можно получить ту же формулу и более строго

Problem B. Building a Stair

- Идея и разработка — Артём Васильев

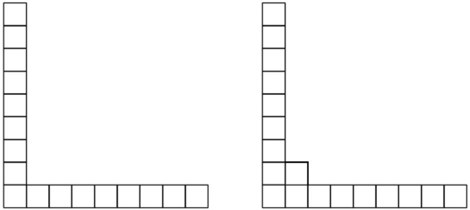
Постановка задачи

- *Лестница* состоит из нескольких башен из кубиков
- Высоты башен слева направо не возрастают
- Лестница *симметрична*, если она симметрична относительно прямой $x = y$
- Построить симметричную лестницу из n кубиков



Решение

- Нечётное n / чётное n



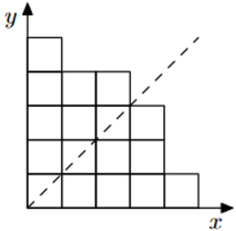
- $n = 2$ — решения нет

Problem C. Counting Stairs

- Идея — Геннадий Короткевич
- Разработка — Артём Васильев

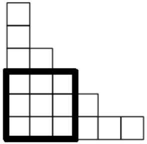
Постановка задачи

- *Лестница* состоит из нескольких башен из кубиков
- Высоты башен слева направо не возрастают
- Лестница *симметрична*, если она симметрична относительно прямой $x = y$
- Посчитать симметричные лестницы из n кубиков



Решение

- Переберём максимальный квадрат $k \times k$ в лестнице. Заметим, что $k \leqslant \sqrt{n}$
- Осталось $n - k^2$ кубиков — чётное число
- Добавим к ответу число способов разбить $\frac{n-k^2}{2}$ кубиков не более чем на k невозрастающих башен



Решение

- $f(i, j)$ — число способов разбить i кубиков не более чем на j невозрастающих башен
- Либо башен менее j, либо все башни содержат хотя бы один кубик
- $f(i, j) = f(i, j-1) + f(i-j, j)$

- Нам интересны $i \leqslant n$ и $j \leqslant \sqrt{n}$
- Асимптотика решения: $\mathcal{O}(n\sqrt{n})$

Problem D. Distinct Substrings

- Идея — Илья Збань
- Подготовка — Павел Маврин

Постановка задачи

- Дана строка p длины $k \leqslant 1000$
- Её повторили много раз и оставили первые $n \leqslant 10^9$ символов, получив строку s
- Сколько различных подстрок в s?

Решение

- Если $p = tt\ldots t$ для некоторой строки t, заменим p на t. Далее будем считать, что у p нет точного периода
- $f(n)$ — число различных подстрок префикса s длины n
- Пусть $n > 2k$. Докажем, что $f(n) = f(n-1) + k$:
  - Пусть t — суффикс s
  - Если $|t| \leqslant n - k$, то t также входит в s на k позиций левее
  - Если $|t| > n - k$, то t входит в s только как суффикс
    - Предположим обратное, пусть t входит в s на x позиций левее. Тогда $\forall i : p[i] = p[(i + x) \bmod k]$. Но $x < k \implies p$ периодична, противоречие
- Для $n \leqslant 2k$ решим задачу как угодно

Решение

- Если $p = tt \ldots t$ для некоторой строки t, заменим p на t. Далее будем считать, что у p нет точного периода
- $f(n)$ — число различных подстрок префикса s длины n
- Пусть $n > 2k$. Докажем, что $f(n) = f(n-1) + k$:
  - Пусть t — суффикс s
  - Если $|t| \leqslant n - k$, то t также входит в s на k позиций левее
  - Если $|t| > n - k$, то t входит в s только как суффикс
    - Предположим обратное, пусть t входит в s на x позиций левее. Тогда $\forall i : p[i] = p[(i + x) \bmod k]$. Но $x < k \implies p$ периодична, противоречие
- Для $n \leqslant 2k$ решим задачу как угодно

Решение

- Если $p = tt\ldots t$ для некоторой строки t, заменим p на t. Далее будем считать, что у p нет точного периода
- $f(n)$ — число различных подстрок префикса s длины n
- Пусть $n > 2k$. Докажем, что $f(n) = f(n-1) + k$:
    - Пусть t — суффикс s
    - Если $|t| \leqslant n - k$, то t также входит в s на k позиций левее
    - Если $|t| > n - k$, то t входит в s только как суффикс
        - Предположим обратное, пусть t входит в s на x позиций левее. Тогда $\forall i : p[i] = p[(i + x) \bmod k]$. Но $x < k \implies p$ периодична, противоречие
- Для $n \leqslant 2k$ решим задачу как угодно

Решение

- Если $p = tt\dots t$ для некоторой строки t, заменим p на t. Далее будем считать, что у p нет точного периода
- $f(n)$ — число различных подстрок префикса s длины n
- Пусть $n > 2k$. Докажем, что $f(n) = f(n-1) + k$:
  - Пусть t — суффикс s
  - Если $|t| \leqslant n - k$, то t также входит в s на k позиций левее
  - Если $|t| > n - k$, то t входит в s только как суффикс
    - Предположим обратное, пусть t входит в s на x позиций левее. Тогда $\forall i : p[i] = p[(i + x) \bmod k]$. Но $x < k \implies p$ периодична, противоречие
- Для $n \leqslant 2k$ решим задачу как угодно

Problem E. Email Destruction

- Идея и разработка — Виталий Аксёнов

Постановка задачи

- Дано k заголовков писем, содержащие в начале несколько “`Re: `”
- Письма образовывали цепочки без пропусков
    - “`hello`”
    - “`Re: hello`”
    - “`Re: Re: hello`”
    - ...
- Некоторые письма удалили, а остальные перемешали
- Могло ли исходно быть n писем?

Решение

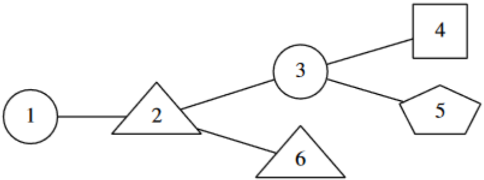
- $m[s]$ — наибольшее число “`Re: `” в одном заголовке письма из цепочки s
- Ограничения малы, можно вычислять $m[s]$ как угодно:
  - Вложенные циклы
  - Ассоциативный массив
  - Хеш-таблица
  - ...
- Просуммируем $m[s] + 1$ по всем различным s и сравним с n

Problem F. Forgotten Land

- Идея и разработка — Илья Збань

Постановка задачи

- Дерево из n городов. В каждом говорят на одном из k языков
- Сложность d_S подмножества городов S — функция от числа языков, на которых говорят либо в городах подмножества, либо на путях между ними
- Найти сумму по всем разбиениям городов на подмножества сумм сложностей всех подмножеств



Решение

- B_k — число способов разбить k городов на подмножества
- $B_0 = 1;\ B_k = \sum\limits_{i=1}^{k} \binom{k}{i} B_{k-i}$
- Переставим местами суммы: ответ равен сумме по подмножествам S величины $d_S \cdot B_{n-|S|}$

Решение

- Переберём маску языков m
- В каких подмножествах городов поддерживаются ровно все языки из маски m?
- Выбросим все города, в которых говорят на других языках
- Города разбились на связные компоненты
- В подмножестве обязаны быть города из одной компоненты
- Рассмотрим связную компоненту размера k. В ней $\binom{k}{i}$ подмножеств размера i. Тогда добавим к ответу $\binom{k}{i} \cdot B_{n-i}$

Что мы упустили?

- *В каких подмножествах городов поддерживаются ровно все языки из маски m?*
- Вместо этого мы учли также подмножества, в которых поддерживаются языки из подмасок m
- $f(m)$ — величина, которую мы по предыдущему алгоритму добавили бы к ответу для маски m
- $g(m)$ — величина, которую нужно было добавить на самом деле
- Тогда $g(m) = f(m) - \sum\limits_{u \subset m, u < m} g(u)$
- Асимптотика решения: $\mathcal{O}(n^2 + n2^k + 3^k)$

Problem G. Generalized German Quotation

- Идея и разработка — Дмитрий Штукенберг

Постановка задачи

- Дана строка из кавычек « и »
- Разбить кавычки на пары, чтобы в каждой паре были разные кавычки и пары образовывали правильную скобочную последовательность

Решение

- Если кавычек левого и правого типа разное число, ответа не существует
- Иначе ответ существует всегда:
  - Найдём две соседние разные кавычки
  - Объединим их в пару
  - Удалим их
  - Будем продолжать, пока строка не станет пустой
- Также существуют другие решения — с использованием стека или динамического программирования по подотрезкам

Problem H. Halves Not Equal

- Идея и разработка — Андрей Станкевич

Постановка задачи

- Известен честный способ разделить золото между двумя людьми в соответствии с их запросами
- Пусть их запросы $a_1 \leqslant a_2$ и нужно разделить сумму $b \leqslant a_1 + a_2$. Тогда они должны получить c_1 и c_2:
  - $b \leqslant a_1 \implies c_1 = c_2 = b/2$
  - $a_1 < b < a_2 \implies c_1 = a_1/2,\ c_2 = b - a_1/2$
  - $a_2 \leqslant b \implies c_1 = a_1/2 + (b - a_2)/2,\ c_2 = a_2/2 + (b - a_1)/2$
- Найти честный способ разделить золото между n людьми, чтобы дележи между всеми парами были честными

Решение

- Если $b > (a_1 + a_2)/2$, разделим вместо этого $a_1 + a_2 - b$, а потом заменим $c_1 := a_1 - c_1$ и $c_2 := a_2 - c_2$ и получим верный ответ
- Корректно обобщается на n человек: если $b > (\sum_{i=1}^{n} a_i)/2$, разделим вместо этого $(\sum_{i=1}^{n} a_i) - b$, а потом заменим $c_i := a_i - c_i$ и получим верный ответ

Решение

- Как для двух человек зависят c_1 и c_2 от b?
  - $b \leqslant a_1 \implies c_1 = c_2 = b/2$
  - $a_1 < b \leqslant (a_1 + a_2)/2 \implies c_1 = a_1/2,\ c_2 = b - a_1/2$
- Увеличиваем и c_1, и c_2 с постоянной скоростью. Когда c_1 достигает $a_1/2$, продолжаем увеличивать только c_2
- Тоже корректно обобщается на n человек!
- Увеличиваем все c_i с постоянной скоростью. Когда c_i достигает $a_i/2$, перестаём увеличивать c_i. Продолжаем, пока сумма c_i не достигнет b

Problem I. Interactive Array Guessing

- Идея и разработка — Борис Минаев

Постановка задачи

- Загаданы $n \leqslant 1000$ массивов различных целых чисел
- Можно сделать не более 20 запросов, каждый о содержимом не более n массивов (необязательно разных)
- В ответ приходит содержимое массивов в запросе, записанное без разделителей между массивами
- Найти содержимое каждого массива

Решение

- Сделаем запрос `? 1 1 2 2 ... n n`
- Как выделить содержимое первого массива?
  - Пользуемся тем, что все элементы массива различны
  - Найдём второе вхождение его первого элемента. Перед этой позицией и будет записан первый массив
- Откинем с начала две копии первого массива и продолжим аналогично искать остальные массивы
- Сделать запрос из $2n$ индексов нельзя, но можно разбить его на два запроса

Problem J. Joined Vessels

- Идея — Виталий Аксёнов
- Разработка — Павел Кунявский

Постановка задачи

- n вертикальных сосудов
- Сосуды i и $i+1$ соединены тонкой трубкой на высоте h_i
- q экспериментов
- Пусть все сосуды изначально пусты. Сколько воды нужно налить в сосуд a, чтобы вода начала появляться в сосуде b?

Решение

- Без ограничения общности, пусть $b < a$
- В сосуде $b+1$ должно быть h_b литров, чтобы вода начала выливаться в сосуд b
- В сосуде $b+2$ должно быть $\max(h_b, h_{b+1})$ литров, чтобы вода начала выливаться в сосуд $b+1$
- $\ldots$
- В сосуде a должно быть $\max(h_b, h_{b+1}, \ldots, h_{a-1})$ литров, чтобы вода начала выливаться в сосуд $a-1$
- В сосудах правее a также будет $x = \max(h_b, h_{b+1}, \ldots, h_{a-1})$ литров — до первого сосуда $i > a$, что $h_{i-1} > x$

Решение

- Обработаем все запросы $b < a$ за $\mathcal{O}(n \log n)$
- Пройдёмся со стеком справа налево, будем поддерживать последовательность рекордных h_i
- Также храним в стеке суммарный объём воды на суффиксах
- Тогда после добавления сосуда b на верху стека будет ответ на запрос (n, b)
- Чтобы получить ответ на запрос (a, b), нужно вычесть объём воды на суффиксе сосудов
- Найдём нужную границу с помощью двоичного поиска

Problem K. Keyboard Consensus

- Идея и разработка — Павел Маврин

Постановка задачи

- Есть два игрока и n клавиатур
- Игроки ходят по очереди, убирая одну клавиатуру из множества
- Известны приоритеты для каждого игрока
- Какая клавиатура будет выбрана в результате?

Решение

- Пусть $n = 2k$ — чётное
- Расположим элементы в следующем порядке:
  - a_1 — наихудший элемент для первого игрока
  - b_1 — наихудший элемент для второго игрока среди оставшихся
  - a_2 — наихудший элемент для первого игрока среди оставшихся
  - b_2 — наихудший элемент для второго игрока среди оставшихся
  - ...
  - a_k — наихудший элемент для первого игрока среди оставшихся
  - b_k — последний оставшийся элемент
- Докажем, что b_k — ответ на задачу

Доказательство (1 из 6)

- Будем доказывать индукцией по k
- Для $k = 1$ ответ очевидно верный
- Заметим, что не существует элемента, который был бы лучше b_k для обоих игроков
- Это следует из того, что каждый предыдущий элемент был худшим для одного из игроков
- Докажем, что оба игрока могут гарантировать себе ответ не хуже b_k

Доказательство (2 из 6)

- Докажем, что первый игрок может гарантировать себе ответ не хуже b_k
- Для этого он делает ход в a_k
- По индукционному предположению, для $k-1$ ответ можно получить с помощью описанной процедуры
- Покажем, что после хода второго игрока ответ будет b_{k-1} или b_k
- Удалим из последовательности $a_1, b_1, a_2, b_2, \ldots, a_k, b_k$ два удаленных элемента и посмотрим, что произойдет, если применить к ней описанную процедуру

Доказательство (3 из 6)

- $a_1, b_1, a_2, b_2, \ldots X \ldots, b_{k-1}, X, b_k$, здесь X — удаленный элемент
- До первого удаленного элемента процедура выберет те же элементы
- Покажем, что вместо удаленного выберется один из двух следующих элементов
  - Пусть, например, удален a_i, тогда вместо него надо выбрать наихудший из оставшихся для первого игрока
  - Заметим, что a_{i+1} — наихудший из всех, кроме b_i
  - Таким образом, процедура выберет a_{i+1} или b_i
- После этого пропуск X сдвинется вправо на одну или две позиции

Доказательство (4 из 6)

- Это продолжается, пока пропуск не доберется до конца последовательности
- $\ldots, X, b_{k-1}, X, b_k$ или $\ldots, X, X, b_k$
- В обоих случаях ответ b_{k-1} или b_k
- Для второго игрока ответ b_k лучше, поэтому он выберет его

Доказательство (5 из 6)

- Теперь докажем, что второй игрок может гарантировать себе ответ не хуже b_k
- Пусть первый игрок сделал ход не в b_k
- Второй делает ход в соседний элемент в последовательности $a_1, b_1, a_2, b_2, \ldots, a_k, b_k$, кроме b_k (такой всегда есть)
- Применим нашу процедуру к оставшимся $n-2$ элементам, убедимся, что ответ будет b_k

Доказательство (6 из 6)

- Пусть первый игрок сделал ход в b_k
- Покажем, в той же технике, что после любого хода второго игрока ответ будет или b_{k-1}, или a_k
- Второй игрок выберет a_k
- Для первого игрока a_k хуже, чем b_k, поэтому делать первый ход в b_k для него не выгодно

Окончание решения

- Если n — нечётное, можно перебрать первый ход и свести задачу к чётному n
- Также можно доказать такой же техникой, что если n — нечётное, то ответ получается той же процедурой, со сменой первого и второго игрока

Problem L. LED-led Paths

- Идея и разработка — Михаил Дворкин

Постановка задачи

- Дан ориентированный ацикличный граф из $n \leqslant 50000$ вершин
- Раскрасить его рёбра в три цвета так, чтобы не существовало одноцветного пути длины более 42

Решение

- Перенумеруем вершины в соответствии с топологической сортировкой
- Решим задачу сразу для полного графа из n вершин, тогда получим решение для любого графа, оставив только нужные рёбра

Решение

- Разобьём вершины на 43 примерно равные группы в порядке номеров. Рёбра между разными группами покрасим в красный цвет. Тогда все красные пути имеют длину до 42
- Каждую группу снова разобьём на 43 примерно равные группы. Рёбра между разными группами покрасим в зелёный цвет. Тогда все зелёные пути имеют длину до 42
- Рёбра внутри последних групп покрасим в синий цвет.
  Так как $43^3 \geqslant n$, все синие пути также имеют длину до 42

Вопросы?