V Открытый кубок школьников

СПбГУ

воскресенье, 19 мая 2013 года

Задача «Магическое выражение»

Задача «Магическое выражение»

- Идея задачи — Илья Малиновский
- Подготовка тестов — Павел Кунявский

Постановка задачи

- Дано корректное арифметическое выражение из знаков +,-,*,(,), цифр и переменной x.
- Необходимо дописать к нему справа что-либо, чтобы получилось корректное выражение, значение которого всегда чётное.

Постановка задачи

- Дано корректное арифметическое выражение из знаков +,-,*,(,), цифр и переменной x.
- Необходимо дописать к нему справа что-либо, чтобы получилось корректное выражение, значение которого всегда чётное.

Решение задачи

- Задача подразумевалась как самая простая в контесте.
- Самым простым решением задачи является "+<выражение из ввода>".
- Ещё есть аналогичное решение "-(<выражение из ввода>)".
- Кроме того, можно было честно посчитать значение выражения при $x = 0$ и $x = 1$ и в зависимости от результата вывести ничего, $+1$, $+x$ или $+x + 1$. Но такое решение значительно сложнее.

Решение задачи

- Задача подразумевалась как самая простая в контесте.
- Самым простым решением задачи является "+<выражение из ввода>".
- Ещё есть аналогичное решение "-(<выражение из ввода>)".
- Кроме того, можно было честно посчитать значение выражения при $x = 0$ и $x = 1$ и в зависимости от результата вывести ничего, $+1$, $+x$ или $+x + 1$. Но такое решение значительно сложнее.

Решение задачи

- Задача подразумевалась как самая простая в контесте.
- Самым простым решением задачи является "+<выражение из ввода>".
- Ещё есть аналогичное решение "-(<выражение из ввода>)".
- Кроме того, можно было честно посчитать значение выражения при $x = 0$ и $x = 1$ и в зависимости от результата вывести ничего, $+1$, $+x$ или $+x + 1$. Но такое решение значительно сложнее.

Задача «Расписание»

Задача «Расписание»

- Идея задачи — Жюри олимпиады
- Подготовка тестов — Павел Кунявский

Постановка задачи

- Дано расписание событий в формате (время, название).
- Также дано несколько запросов: для момента времени определить, какое событие среди начинающихся не раньше этого момента начинается раньше всех.

Постановка задачи

- Дано расписание событий в формате (время, название).
- Также дано несколько запросов: для момента времени определить, какое событие среди начинающихся не раньше этого момента начинается раньше всех.

Решение задачи

- Ограничения в этой задаче такие, что позволяют ответ на каждый запрос находить линейным проходом по массиву событий.

Задача «Священные овцы»

Задача «Священные овцы»

- Идея задачи — Илья Малиновский
- Подготовка тестов — Илья Малиновский

Постановка задачи

- Рассмотрим точку с целочисленными координатами на оси Ox. Назовём <u>галочкой</u> фигуру, образованную двумя лучами, выходящими из этой точки в верхнюю полуплоскость и образующими равные углы с положительным и отрицательным направлениями оси Ox, соответственно. Точку на оси Ox назовём <u>основанием</u> галочки.
- Даны две точки с целочисленными координатами.
- Найти основания всех галочек, содержащих обе данные точки.

Постановка задачи

- Рассмотрим точку с целочисленными координатами на оси Ox. Назовём галочкой фигуру, образованную двумя лучами, выходящими из этой точки в верхнюю полуплоскость и образующими равные углы с положительным и отрицательным направлениями оси Ox, соответственно. Точку на оси Ox назовём основанием галочки.
- Даны две точки с целочисленными координатами.
- Найти основания всех галочек, содержащих обе данные точки.

Постановка задачи

- Рассмотрим точку с целочисленными координатами на оси Ox. Назовём галочкой фигуру, образованную двумя лучами, выходящими из этой точки в верхнюю полуплоскость и образующими равные углы с положительным и отрицательным направлениями оси Ox, соответственно. Точку на оси Ox назовём основанием галочки.
- Даны две точки с целочисленными координатами.
- Найти основания всех галочек, содержащих обе данные точки.

Решение задачи (начало)

- Вначале проверим, что обе точки не ниже оси Ox и хотя бы одна над осью (в противном случае ответ 0)
- Если одна из точек на оси Ox, то она и есть единственный ответ.
- Если у точек разные координаты y, то проведём через них прямую и найдём точку пересечения этой прямой с осью Ox. Если её координата x целая, точка входит в ответ. Тем самым, разобран случай *обе точки лежат на одной стороне галочки*.

Решение задачи (начало)

- Вначале проверим, что обе точки не ниже оси Ox и хотя бы одна над осью (в противном случае ответ 0)
- Если одна из точек на оси Ox, то она и есть единственный ответ.
- Если у точек разные координаты y, то проведём через них прямую и найдём точку пересечения этой прямой с осью Ox. Если её координата x целая, точка входит в ответ. Тем самым, разобран случай *обе точки лежат на одной стороне галочки*.

Решение задачи (начало)

- Вначале проверим, что обе точки не ниже оси Ox и хотя бы одна над осью (в противном случае ответ 0)
- Если одна из точек на оси Ox, то она и есть единственный ответ.
- Если у точек разные координаты y, то проведём через них прямую и найдём точку пересечения этой прямой с осью Ox. Если её координата x целая, точка входит в ответ. Тем самым, разобран случай *обе точки лежат на одной стороне галочки*.

Решение задачи (окончание)

- Если у точек разные координаты x, то опустим из них перпендикуляры на x. Теперь предположим, что искомая точка уже найдена. Тогда треугольники, образованные этой точкой и парами (исходная точка, основание перпендикуляра), будут подобны по двум углам. Записав пропорцию для соответствующих сторон, получим выражение для вычисления неизвестной координаты искомой точки.
- Случаи закончились :) При решении следовало либо производить все вычисления в целых числах, либо аккуратно подбирать *eps* при сравнении

Решение задачи (окончание)

- Если у точек разные координаты x, то опустим из них перпендикуляры на x. Теперь предположим, что искомая точка уже найдена. Тогда треугольники, образованные этой точкой и парами (исходная точка, основание перпендикуляра), будут подобны по двум углам. Записав пропорцию для соответствующих сторон, получим выражение для вычисления неизвестной координаты искомой точки.
- Случаи закончились :) При решении следовало либо производить все вычисления в целых числах, либо аккуратно подбирать *eps* при сравнении

Задача «Кварт»

Задача «Кварт»

- Идея задачи — Николай Карпов
- Подготовка тестов — Николай Карпов

Постановка задачи

- Дано натуральное число n.
- Рассмотрим все перестановки из n элементов.
- Для каждой перестановки π посчитаем, для скольких i выполняется $\pi(i) = i$.
- Требуется найти остаток от деления суммы найденных величин на 43.

Постановка задачи

- Дано натуральное число n.
- Рассмотрим все перестановки из n элементов.
- Для каждой перестановки π посчитаем, для скольких i выполняется $\pi(i) = i$.
- Требуется найти остаток от деления суммы найденных величин на 43.

Постановка задачи

- Дано натуральное число n.
- Рассмотрим все перестановки из n элементов.
- Для каждой перестановки π посчитаем, для скольких i выполняется $\pi(i) = i$.
- Требуется найти остаток от деления суммы найденных величин на 43.

Постановка задачи

- Дано натуральное число n.
- Рассмотрим все перестановки из n элементов.
- Для каждой перестановки π посчитаем, для скольких i выполняется $\pi(i) = i$.
- Требуется найти остаток от деления суммы найденных величин на 43.

Решение задачи

- Посчитаем отдельно вклад каждого элемента перестановки в сумму.
- Число стоит на своей позиции ровно в $(n-1)!$ перестановках, так как мы должны зафиксировать это число и можем как угодно переставлять остальные. Значит, его вклад в сумму равен $(n-1)!$.
- Всего в перестановке n элементов. Значит, суммарный вклад равен $(n-1)! \times n = n!$. То есть, требуется лишь вывести остаток от деления $n!$ на 43.

Решение задачи

- Посчитаем отдельно вклад каждого элемента перестановки в сумму.
- Число стоит на своей позиции ровно в $(n-1)!$ перестановках, так как мы должны зафиксировать это число и можем как угодно переставлять остальные. Значит, его вклад в сумму равен $(n-1)!$.
- Всего в перестановке n элементов. Значит, суммарный вклад равен $(n-1)! \times n = n!$. То есть, требуется лишь вывести остаток от деления $n!$ на 43.

Решение задачи

- Посчитаем отдельно вклад каждого элемента перестановки в сумму.
- Число стоит на своей позиции ровно в $(n-1)!$ перестановках, так как мы должны зафиксировать это число и можем как угодно переставлять остальные. Значит, его вклад в сумму равен $(n-1)!$.
- Всего в перестановке n элементов. Значит, суммарный вклад равен $(n-1)! \times n = n!$. То есть, требуется лишь вывести остаток от деления $n!$ на 43.

Решение задачи

При вычислении $n!$ удобно каждый раз после умножения брать результат по модулю 43, чтобы избежать переполнения.

Задача «Архелон»

Задача «Архелон»

- Идея задачи — Ольга Бурсиан
- Подготовка тестов — Ольга Бурсиан

Постановка задачи

- Черепашка умеет за один ход делать один шаг вперед, поворачиваться налево на 60° и направо тоже на 60°. Дана строка, которой закодирован путь черепашки.
- Найти количество точек, в которых черепашка пересекала свой собственный путь.

Постановка задачи

- Черепашка умеет за один ход делать один шаг вперед, поворачиваться налево на 60° и направо тоже на 60°. Дана строка, которой закодирован путь черепашки.
- Найти количество точек, в которых черепашка пересекала свой собственный путь.

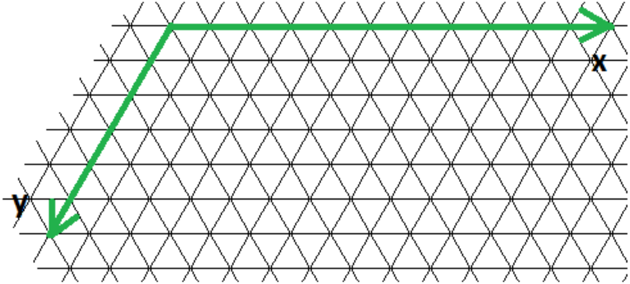
Первое решение

Рассмотрим плоскость, на которой проведены прямые:

1. $y = \frac{\sqrt{3}}{2} \cdot i,$
2. $y = \sqrt{3} \cdot x + \sqrt{3} \cdot i,$
3. $y = -\sqrt{3} \cdot x + \sqrt{3} \cdot i$

при всех $i = \ldots, -1, 0, 1, 2, 3, \ldots$. Мы можем считать, что черепашка находится в точке с кооординатами $(0, 0)$ в начальный момент времени и после первого шага окажется в точке с координатами $(1, 0)$. Заметим, что тогда все движение Архелона происходит на решетке, образованной проведенными прямыми.

Первое решение



Первое решение

На этой плоскости можно ввести декартову систему координат так, что абсциссой будет прямая $y = 0$, а ординатой — $y = -\sqrt{3} \cdot x$. Тогда движение черепашки происходит в обычной декартовой системе координат, только оси пересекаются не под прямым углом. Таким образом, движение черепашки сводится к движению по ребрам графа.

Первое решение

Заведем переменные, в которых хранится положение черепашки и направление ее движения. Заведем массив d всех возможных направлений движения черепашки в порядке обхода по часовой стрелке. В нем хранятся вектора $(1, 0)$, $(1, 1)$, $(0, 1)$, $(-1, 0)$, $(-1, -1)$, $(0, -1)$. Пусть текущее положение черепашки — вектор (x, y), целая переменная b указывает номер в списке векторов направления движения, куда сейчас смотрит голова черепашки. Читая строку входного файла, меняем их соответствующим образом.

Первое решение

Если дан символ L или R, то изменяем b, если F — то вектор (x, y) меняется на $(x, y) + (dx, dy)$, где (dx, dy) — текущий вектор направления, взятый из массива d. При этом нужно проверить, не были ли мы уже в этой точке. Все точки пути черепашки кладем в массив, список или другую структуру данных, в которой поиск выполняется быстрее. При данных ограничениях это неважно. Если оказалось, что в этой точке мы уже были, то нужно определить, это точка самокасания или самопересечения и отдельно рассмотреть случаи, когда это первая или последняя точка.

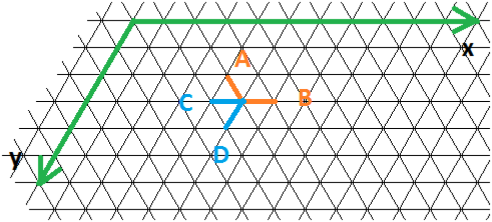
Как определить, что точка является точкой самопересечения (I способ)

Пусть OA, OB, OC, OD - 4 вектора, исходящие из той точки, которую мы проверяем, OA, OB образованы отрезками кривой с номерами i и $i+1$, а OC и OD — отрезками с номерами j и $j+1$. Они все содержатся в массиве d.

Как определить, что точка является точкой самопересечения (I способ)

Если есть такой элемент в массиве d, что начиная с него и идя в порядке обхода по часовой стрелке или против (после последнего элемента снова идем на первый), встретим сначала пару OA и OB, а потом пару OC и OD, то значит, это точка, в которой кривая касается сама себя и она не подходит. Если же такого элемента в массиве d не найдется, то точка является точкой самопересечения.

Как определить, что точка является точкой самопересечения (II способ)



Пусть точка является точкой самокасания. Тогда хотя бы для одного из векторов выполнено некоторое условие. Пусть это вектор OA.

Как определить, что точка является точкой самопересечения (II способ)

Тогда оно выражается в том, что вектора OC и OD лежат по одну сторону от прямой, проведенный через вектор OA. Если при этом вектор OB лежит по другую сторону от этой прямой, то точка является точкой самокасания. Если это не так, но при этом вектора OC и OD лежат по одну сторону от прямой, проведенной через вектор OB, то точка тоже является точкой самокасания.

Второе решение

Второе решение чисто геометрическое. Также храним текущие координаты черепашки и текущий вектор направления. При повороте на угол α вектор (x, y) умножается на матрицу $\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}$, то есть превращается в $(x * \cos\alpha + y * (-\sin\alpha), x * \sin\alpha + y * \cos\alpha)$. В этом решении появляется погрешность при сравнении координат на равенство, но при заданных ограничениях решение с погрешностью тоже проходит.

Задача «Команды»

Задача «Команды»

- Идея задачи — Жюри олимпиады
- Подготовка тестов — Николай Карпов

Постановка задачи

Есть список разбиения команд на кучки, требуется продолжить его как можно дальше.

Оценка на ответ

Заметим, что оценка сверху на ответ 9. Это следует хотя бы из того, что с одним человеком может сыграть только 9 различных пар участников из двух других команд.

Решение

Перебор.

- Перебираем, какое разбиение сделаем
- Проверяем, что оно подходит под условия
- Если сделали разбиение длиной 9, то выходим из перебора

Почему быстро работает

Почему быстро работает.
Максимальные наборы по включению бывает величины 5, 7, 9.

- Если ответ 9, то перебор быстро находит ответ.
- Ответ 7 возможен, только когда во входном файле задано не менее 4 разбиений на кучки.
- Ответ 5 возможен, только когда во входном файле задано все пять разбиений.

Таким образом, перебор быстро находит ответ 9 и выходит, либо глубина перебора маленькая

Задача «Сокровища в книгах»

Задача «Сокровища в книгах»

- Идея задачи — Николай Карпов
- Подготовка тестов — Павел Кунявский

Постановка задачи

- В условии описана строка Туэ-Морса.
- Необходимо найти наибольшую общую подстроку, начинающуюся в двух заданных позициях.

Постановка задачи

- В условии описана строка Туэ-Морса.
- Необходимо найти наибольшую общую подстроку, начинающуюся в двух заданных позициях.

Решение задачи

- Будем решать задачу рекурсивно.
- База. Если числа A и B имеют разную чётность чила бит, то $Ans(A, B) = 0$.
- Если среди чисел A и B есть нечётное, то $Ans(A, B) = 1 + Ans(A + 1, B + 1)$.
- Если числа A и B оба чётные, то $Ans(A, B) = 2 \cdot Ans(\frac{A}{2}, \frac{B}{2})$

Решение задачи

- Будем решать задачу рекурсивно.
- База. Если числа A и B имеют разную чётность чила бит, то $Ans(A, B) = 0$.
- Если среди чисел A и B есть нечётное, то $Ans(A, B) = 1 + Ans(A + 1, B + 1)$.
- Если числа A и B оба чётные, то $Ans(A, B) = 2 \cdot Ans(\frac{A}{2}, \frac{B}{2})$

Решение задачи

- Будем решать задачу рекурсивно.
- База. Если числа A и B имеют разную чётность чила бит, то $Ans(A, B) = 0$.
- Если среди чисел A и B есть нечётное, то $Ans(A, B) = 1 + Ans(A + 1, B + 1)$.
- Если числа A и B оба чётные, то $Ans(A, B) = 2 \cdot Ans(\frac{A}{2}, \frac{B}{2})$

Доказательство

- База и первый переход очевидны.
- Если оба числа чётные, то позиции разбиваются на пары соседних. При этом, если $s[i] = S$, то $s[2i..2i+1] = SL$, а иначе $s[2i..2i+1] = LS$.
- Пока числа A, B одинаковой чётности, может случиться не более $2 \log n$ шагов.
- Когда числа A и B стали разной чётности, то быстро найдётся момент, когда при прибавлении 1, чётность изменится у одного из двух.

Доказательство

- База и первый переход очевидны.
- Если оба числа чётные, то позиции разбиваются на пары соседних. При этом, если $s[i] = S$, то $s[2i..2i+1] = SL$, а иначе $s[2i..2i+1] = LS$.
- Пока числа A, B одинаковой чётности, может случиться не более $2\log n$ шагов.
- Когда числа A и B стали разной чётности, то быстро найдётся момент, когда при прибавлении 1, чётность изменится у одного из двух.

Доказательство

- База и первый переход очевидны.
- Если оба числа чётные, то позиции разбиваются на пары соседних. При этом, если $s[i] = S$, то $s[2i..2i+1] = SL$, а иначе $s[2i..2i+1] = LS$.
- Пока числа A, B одинаковой чётности, может случиться не более $2 \log n$ шагов.
- Когда числа A и B стали разной чётности, то быстро найдётся момент, когда при прибавлении 1, чётность изменится у одного из двух.

Ещё одно решение задачи

- Задачу можно решать бинарным поиском, если научиться сравнивать две подстроки
- Две подстроки можно сравнивать хешами, для этого нужно уметь считать хеш префикса.
- Хеш префикса считается рекурсивно, если заметить, что строка получается приписыванием к себе строки S с заменой символа на противоположный.

Ещё одно решение задачи

- Задачу можно решать бинарным поиском, если научиться сравнивать две подстроки
- Две подстроки можно сравнивать хешами, для этого нужно уметь считать хеш префикса.
- Хеш префикса считается рекурсивно, если заметить, что строка получается приписыванием к себе строки S с заменой символа на противоположный.

Ещё одно решение задачи

- Задачу можно решать бинарным поиском, если научиться сравнивать две подстроки
- Две подстроки можно сравнивать хешами, для этого нужно уметь считать хеш префикса.
- Хеш префикса считается рекурсивно, если заметить, что строка получается приписыванием к себе строки S с заменой символа на противоположный.

Задача «Красная свадьба»

Задача «Красная свадьба»

- Идея задачи — Илья Малиновский
- Подготовка тестов — Илья Малиновский

Постановка задачи

- Даны два множества, в каждом из которых по n элементов, пронумерованных от 1 до n.
- Разрешено удалять один элемент из любой доли, а также пары элементов из разных долей, номера которых отличаются не более, чем на k.
- Для каждой возможной операции удаления известна её стоимость.
- Требуется найти минимальную стоимость удаления всех элементов.

Постановка задачи

- Даны два множества, в каждом из которых по n элементов, пронумерованных от 1 до n.
- Разрешено удалять один элемент из любой доли, а также пары элементов из разных долей, номера которых отличаются не более, чем на k.
- Для каждой возможной операции удаления известна её стоимость.
- Требуется найти минимальную стоимость удаления всех элементов.

Постановка задачи

- Даны два множества, в каждом из которых по n элементов, пронумерованных от 1 до n.
- Разрешено удалять один элемент из любой доли, а также пары элементов из разных долей, номера которых отличаются не более, чем на k.
- Для каждой возможной операции удаления известна её стоимость.
- Требуется найти минимальную стоимость удаления всех элементов.

Постановка задачи

- Даны два множества, в каждом из которых по n элементов, пронумерованных от 1 до n.
- Разрешено удалять один элемент из любой доли, а также пары элементов из разных долей, номера которых отличаются не более, чем на k.
- Для каждой возможной операции удаления известна её стоимость.
- Требуется найти минимальную стоимость удаления всех элементов.

Решение номер 1

- Автор идеи — Егор Суворов
- Построим двудольный граф: доли — множества, вершины — элементы, рёбра — пары элементов, которые можно взять одновременно. Вес ребра — стоимость взятия соответствующей пары.
- Построим на этом двудольном графе сеть. Пропускные способности всех рёбер по 1.
- Запустим обычный min-cost-flow. При помощи него найдём потоки минимальной стоимости для всех величин от 0 до n. Скажем, что те вершины, у которых не нашлось инцидентного им насыщенного ребра, будут взяты по одной.

Решение номер 1

- Автор идеи — Егор Суворов
- Построим двудольный граф: доли — множества, вершины — элементы, рёбра — пары элементов, которые можно взять одновременно. Вес ребра — стоимость взятия соответствующей пары.
- Построим на этом двудольном графе сеть. Пропускные способности всех рёбер по 1.
- Запустим обычный min-cost-flow. При помощи него найдём потоки минимальной стоимости для всех величин от 0 до n. Скажем, что те вершины, у которых не нашлось инцидентного им насыщенного ребра, будут взяты по одной.

Решение номер 1

- Автор идеи — Егор Суворов
- Построим двудольный граф: доли — множества, вершины — элементы, рёбра — пары элементов, которые можно взять одновременно. Вес ребра — стоимость взятия соответствующей пары.
- Построим на этом двудольном графе сеть. Пропускные способности всех рёбер по 1.
- Запустим обычный min-cost-flow. При помощи него найдём потоки минимальной стоимости для всех величин от 0 до n. Скажем, что те вершины, у которых не нашлось инцидентного им насыщенного ребра, будут взяты по одной.

Решение номер 1

- Автор идеи — Егор Суворов
- Построим двудольный граф: доли — множества, вершины — элементы, рёбра — пары элементов, которые можно взять одновременно. Вес ребра — стоимость взятия соответствующей пары.
- Построим на этом двудольном графе сеть. Пропускные способности всех рёбер по 1.
- Запустим обычный min-cost-flow. При помощи него найдём потоки минимальной стоимости для всех величин от 0 до n. Скажем, что те вершины, у которых не нашлось инцидентного им насыщенного ребра, будут взяты по одной.

Решение номер 1

- Теперь мы знаем минимальную стоимость удаления всех элементов при условии, что ровно $2 * i$ из них составляют пары, для каждого i от 0 до n. Минимум из этих величин и есть ответ.

Решение номер 2

- Автор идеи — Илья Малиновский
- Будем удалять элементы в таком порядке, чтобы в любой момент из первой доли были удалены элементы с 1-го по t-й включительно (t произвольное) и только они, а из второй доли — все элементы с 1-го по $t - k - 1$-й включительно и, возможно, какие-то элементы с $t - k$-го по $t + k$-й включительно.

Решение номер 2

- Автор идеи — Илья Малиновский
- Будем удалять элементы в таком порядке, чтобы в любой момент из первой доли были удалены элементы с 1-го по t-й включительно (t произвольное) и только они, а из второй доли — все элементы с 1-го по $t-k-1$-й включительно и, возможно, какие-то элементы с $t-k$-го по $t+k$-й включительно.

Решение номер 2

- Зададим текущее состояние числом t и битовой маской для обозначения взятых элементов из отрезка $[t-k;\ t+k]$.
- Задача решается при помощи динамики с описанным выше состоянием.

Решение номер 2

- Зададим текущее состояние числом t и битовой маской для обозначения взятых элементов из отрезка $[t - k; t + k]$.
- Задача решается при помощи динамики с описанным выше состоянием.

Решение номер 2

- Зададим текущее состояние числом t и битовой маской для обозначения взятых элементов из отрезка $[t - k;\ t + k]$.
- Задача решается при помощи динамики с описанным выше состоянием.

Решение номер 2

- Зададим текущее состояние числом t и битовой маской для обозначения взятых элементов из отрезка $[t - k; t + k]$.
- Задача решается при помощи динамики с описанным выше состоянием.

Задача «Прямоугольники»

Задача «Прямоугольники»

- Идея задачи — Павел Кунявский
- Подготовка тестов — Павел Кунявский

Постановка задачи

Вам надо научиться отвечать на запрос о пересечении прямоугольников в подтаблице.

Решение

- Предподсчёт за $O(n^2 \log^2 n)$
- Ответ на запрос за $O(1)$

Предподсчёт

- Посчитаем пересечение всех прямоугольников с верхним левым углом в точке (i, j) и размера $2^k \times 2^l$. Обозначим это пересечение за $\mathrm{I}(i, j, k, l)$
- Величина $\mathrm{I}(i, j, k, l)$ легко пересчитывается через $\mathrm{I}(i', j', k-1, l)$ и $\mathrm{I}(i', j', k, l-1)$
- Предподсчитываем всё в порядке увеличения l, k

Предподсчёт

- Посчитаем пересечение всех прямоугольников с верхним левым углом в точке (i, j) и размера $2^k \times 2^l$. Обозначим это пересечение за $\mathrm{I}(i, j, k, l)$
- Величина $\mathrm{I}(i, j, k, l)$ легко пересчитывается через $\mathrm{I}(i', j', k-1, l)$ и $\mathrm{I}(i', j', k, l-1)$
- Предподсчитываем всё в порядке увеличения l, k

Предподсчёт

- Посчитаем пересечение всех прямоугольников с верхним левым углом в точке (i, j) и размера $2^k \times 2^l$. Обозначим это пересечение за $\mathrm{I}(i, j, k, l)$
- Величина $\mathrm{I}(i, j, k, l)$ легко пересчитывается через $\mathrm{I}(i', j', k-1, l)$ и $\mathrm{I}(i', j', k, l-1)$
- Предподсчитываем всё в порядке увеличения l, k

Ответ на запрос

- Находим такое максимальное k и l, что подтаблица таких размеров влезает в нашу подтаблицу
- Выражаем ответ через четыре прямоугольника: $\mathrm{I}(r_1, c_1, k, l)$, $\mathrm{I}(r_2 - 2^k, c_1, k, l)$, $\mathrm{I}(r_1, c_2 - 2^l, k, l)$, $\mathrm{I}(r_2 - 2^k, c_2 - 2^l, k, l)$
- Для асимптотики $O(1)$ надо предподсчитать такие максимальные k, l для всех возможных длин

Ответ на запрос

- Находим такое максимальное k и l, что подтаблица таких размеров влезает в нашу подтаблицу
- Выражаем ответ через четыре прямоугольника: $\mathrm{I}(r_1, c_1, k, l)$, $\mathrm{I}(r_2 - 2^k, c_1, k, l)$, $\mathrm{I}(r_1, c_2 - 2^l, k, l)$, $\mathrm{I}(r_2 - 2^k, c_2 - 2^l, k, l)$
- Для асимптотики $O(1)$ надо предподсчитать такие максимальные k, l для всех возможных длин

Ответ на запрос

- Находим такое максимальное k и l, что подтаблица таких размеров влезает в нашу подтаблицу
- Выражаем ответ через четыре прямоугольника: $\mathrm{I}(r_1, c_1, k, l)$, $\mathrm{I}(r_2 - 2^k, c_1, k, l)$, $\mathrm{I}(r_1, c_2 - 2^l, k, l)$, $\mathrm{I}(r_2 - 2^k, c_2 - 2^l, k, l)$
- Для асимптотики $O(1)$ надо предподсчитать такие максимальные k, l для всех возможных длин

Задача «Партии»

Задача «Партии»

- Идея задачи — Николай Карпов
- Подготовка тестов — Николай Карпов

Постановка задачи

У вас есть постепенно растущее семейство множеств. Вам нужно для каждого нового добавляемого множества отвечать сколько таких множество уже есть в этом семействе.

Вспомогательная лемма

Пускай у нас есть вектор $x \in \{0,1\}^n$
$Pr_{v \in \{0,1\}^n}(x * v = 0) = \frac{1}{2}$ если $x \neq \{0\}^n$.

Выберем 128 случайных векторов v, а x будет характерестическими функциями множеств. По сути мы можем составить функцию h из $1 \ldots 10^6$ в числа $[0, 2^{128} - 1)$, сопоставив каждому числу случайное число.

Тогда если $H(S) = \bigoplus_{iinS} h(i)$. То вероятность, что два множества совпадут $H(T) = H(S)$ не более 2^{-128}. Это следует из леммы.
Так как пар множеств всего $4 \cdot 10^{12}$ то вероятность, что мы ошибёмся хоть один раз мала.

Заметим, что $H(S \oplus T) = H(S) \oplus H(T)$. Поэтому всё легко пересчитывается. В итоге можно посчитать все хэши, потом отсортировать их, и посчитать ответ. Получаем решение за $O(n \log n)$ с маленькой константой.

EOF