Задача A. 11-45-G: Количество цифр

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: математика

Согласно ограничениям задачи $0 \leqslant n \leqslant 9$. А это значит, что для любого теста ответ равен 1.

Задача B. XBOT 4000: Мячи

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: математика

Пусть мячи переложить можно. Тогда для некоторого целого x надо получить состояние (x, x) такое, что $2 \cdot x = n + m$. Такое x существует, если $n + m$ — чётное число.

Задача C. K-VRC: Перестановка

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: линейный массив

Создадим массив w, изначально заполненный нулями. Теперь для всех i заполним этот массив $w_{p_i} = i$. В качестве ответа выведем содержимое массива.

Задача D. 11-45-G: Прогулка робота

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: манхэттенское расстояние

Сначала очевидное свойство: во время прохода от x_{min} до x_{max} или наоборот мы посетим все остальные обязательные точки x. То есть мы можем за $x_{max} - x_{min}$ шагов, посетить обязательные точки, что является минимальным возможным количеством ходов. Так же и для y, и мы можем делать это независимо от x. Ответ равен $x_{max} - x_{min} + y_{max} - y_{min}$.

Задача E. XBOT 4000: Соседние цвета

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: жадный алгоритм

Выгоднее всего выводить символы, чередуя их типы. Так, если количество вхождений a больше количества вхождений b, то выгодно выводить ab $count_b$ раз и далее букву a $count_a - count_b$ раз. И наоборот, если количество вхождений b больше количества вхождений a.

Задача F. K-VRC: Делимое не делимое

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: наибольший общий делитель, алгоритм Евклида

Если каждое число множества делится на x, то НОД этого множества также делится на x. То есть мы можем найти число g — НОД чисел массива a. И за линейное время проверить для каждого i, что g делится на b_i.

Задача G. XBOT 4000: Пары родственников

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: математика, двоичная арифметика, перебор

Заметим, что количество подходящих чисел около 200. Мы можем перебирать, чему равно одно число и чему будет равна сумма, тогда мы будем точно знать второе число. Таким образом мы сможем посчитать ответ за $O(200 \cdot n)$.

Задача H. XBOT 4000: Очередь

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: жадные алгоритмы

Давайте воспользуемся следующим жадным алгоритмом: отсортируем котов по времени. Будем идти от меньшего к большему, если текущее время не более чем время кота, то будем брать этого кота, иначе не будем брать этого кота (формально мы оставим его на самый конец, чтоб он не ухудшил ответ). Количество взятых котов будет максимальным количеством довольных котов, а ответ будет равен n минус данное число.

Задача I. 11-45-G: Функции

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: математика, двоичная арифметика

Введём функцию $c(x) = \sum_{i=1}^{x} f(i)$. Ответ на запрос равен $c(r) - c(l-1)$. Теперь наша задача — научиться быстро считать $c(x)$. Посчитаем, сколько единичных битов есть в числе x. Теперь нам интересны меньшие числа. Будем перебирать единичный бит, который впервые будет меньше в нашем меньшем числе x. Там оставим 0. Пусть это был бит под номером i. Тогда есть 2^i чисел с данным префиксом. Нужно посчитать, сколько будет единиц среди них. Количество единиц равно $i \cdot 2^{i-1}$. То есть для i ответ равен $suf_i \cdot 2^i + i \cdot 2^{i-1}$.

Задача J. 11-45-G: Swap на строках

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: математика, префиксные суммы, структуры данных

Легко показать, что с запросом из условия мы можем добавить также запросы вида: (i, j) поменять a_i, a_j и i, j поменять b_i, b_j.

```
(1, 2); (2, 2); (1, 2)

ab|db|da|ba
cd|ca|cb|cd
```

То есть $f(a, b) = 1$, если каждая буква встречается чётное количество раз суммарно в строках a и b. Теперь будем решать задачу с префиксными суммами. $pref_{i,j} = 0..1$ — чётность вхождения символа j на интервале $1..i$. Наша задача — узнать количество пар (x, y) таких, что $pref_x = pref_y$.

Это можно сделать за $O(26 \cdot n \cdot \log(n))$ при помощи ассоциативного массива.

Задача K. K-VRC: Пары

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: перебор, оптимизация, структуры данных

Сначала решим задачу перебором. Будем перебирать, чему равна сумма пары s от 1 до $2 \cdot \max(a_i)$. Зная s можно посчитать, сколько существует i таких, что $s - a_i$ также есть в массиве. Количество пар равно количеству данных индексов i, делённому на 2. Асимптотика — $O(\max(a_i) \cdot n)$.

Теперь будем оптимизировать решение. Можно перебирать не сумму, а медиану пары mid. Будем считать количество индексов i таких, что $a_i \leqslant mid$ и что в массиве есть число a_j $(mid - a_i = a_j - mid)$. Это удобно реализовать при помощи битсетов.

Пусть есть битсет a такой, что $a_i = 1$, если в массиве есть число $mid - i$, и битсет b такой, что $b_i = 1$, если в массиве есть число $mid + i$. Количество индексов равно $(a \ \& \ b).count()$. Это можно легко считать, фиксируя $mid = 1$ и постепенно увеличивая его.

Итоговая асимптотика: $O(\frac{max(a_i) \cdot n}{64})$

Задача L. 11-45-G: Запросы на массиве

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: сложные структуры данных

Будем отвечать на запросы *offline*, перемещая r вправо и сохраняя ответ для каждой левой точки в дереве отрезков. Дерево будет поддерживать две операции: установить минимум на отрезке и получить значение в точке. Предположим, что среди двух элементов a_i и a_j в нашем массиве $a_i > a_j$ и $i < j$ и решим задачу дважды — для исходного массива и для реверсированного.

Рассмотрим один шаг перемещения r и добавления нового элемента x в позицию i. Находим первый элемент слева от i, который не меньше x, обозначим его через $a_j = y$. Теперь ответ для всех левых конечных точек в диапазоне $[0, j]$ равен $y - x$. Далее находим некоторое $a'_j = y'$ такое, что $x \leqslant y' < y$ и $j' < j$, и ответ для всех левых конечных точек в диапазоне $[0, j']$ не превосходит $y' - x$. Также давайте выбирать j' так, что $y - y' > y' - x$. Заметим, что значение $y' - x$ будет уменьшаться в два раза и таким образом мы сделаем не более $\log_2(10^9)$ операций.

Итоговая асимптотика: $O(n \cdot \log_2(n) \cdot \log_2(10^9) + q \cdot \log_2(n))$.