Задача A. Борис и решение уравнения

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

Заметим, что число a может принимать значения от 1 до $n-1$, а b равно $n-a$. Количество возможных пар: $n-1$.

Задача B. Борис, Петя и карточки

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: реализация, подсчёт, математика, битовые операции

Существует несколько способов решения этой задачи. Рассмотрим три из них:

1. Воспользоваться массивом подсчёта, и найти число, которое встречается один раз.
2. Ответ будет равен $2 \cdot \sum_{i=1}^{n}(i) - \sum_{i=1}^{2 \cdot n-1}(a_i)$.
3. Ответ равен битовому исключающему ИЛИ всех чисел в массиве.

Задача C. Борис и сейф

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: реализация, двумерные массивы

Для i-го столбца матрицы a будем проверять, есть ли в нём число b_i. Если найдётся столбец, в котором нет нужного числа, то ответ «`No`», иначе — «`Yes`».

Задача D. Борис и подарок

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

Одним из решений является вывод первых n нечётных чисел. Например: $1+3 = 2^2$, $1+3+5 = 3^2$, $1+3+5+7 = 4^2$. Последним числом в этой последовательности будет $2 \cdot n - 1$, что подходит для ограничений задачи.

Задача E. Борис и утерянный пароль

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: реализация, поиск в строке

В этой задаче для каждой строки ввода надо выяснить, есть ли в ней искомая подстрока. Например, в C++ это можно сделать с помощью функции `string::find()`.

Задача F. Борис и НОК

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: математика, НОК, НОД

Вспомним, как находить НОК двух чисел:

$\text{НОК}(a, b) = \frac{a \cdot b}{\text{НОД}(a,b)}$.

Предположим, что все НОД = 1, тогда наиболее выгодной парой будет $(r-1, r)$.

Решим задачу без указанного предположения: вспомним, что соседние натуральные числа — взаимно просты, и максимальное НОК будет равно $(r-1) \cdot r$.

Задача G. Борис и пицца

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: линейное динамическое программирование, жадные алгоритмы

Будем вычислять dp_i — за какое минимальное количество пицц нужно заплатить, чтобы угостить всех гостей в первые i дней.

В $i-$й день можно купить либо a_i пицц, либо $a_i + a_{i+1}$ пицц. Ответ будет находиться в dp_n.

Также эту задачу можно решить с помощью жадного алгоритма.

Задача H. Борис и среднее арифметическое

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

Будем решать задачу поэтапно. Сначала заметим, что компоненты с одинаковым знаменателем можно объединить в одну дробь.

Будем перебирать len и добавлять к ответу $\frac{\sum_{i=1}^{n}(a_i \cdot b_{len,i})}{len}$, где $b_{len,i}$ — количество подотрезков длиной len таких, что i находится в этом подотрезке.

Исследуем закономерности: переберём len и оценим значения b.

Для $n = 5$:

$len = 1$: $b =$ `[1, 1, 1, 1, 1]`.
$len = 2$: $b =$ `[1, 2, 2, 2, 1]`.
$len = 3$: $b =$ `[1, 2, 3, 2, 1]`.
$len = 4$: $b =$ `[1, 2, 2, 2, 1]`.
$len = 5$: $b =$ `[1, 1, 1, 1, 1]`.

При переходе от len к $len + 1$ будем считать $\sum_{i=1}^{n}(a_i \cdot b_{len,i})$, в зависимости от len добавляя или отнимая от ответа сумму некоторого подотрезка.

Итоговая асимптотика решения: $O(n)$.

Задача I. Борис и трубопровод

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: n-мерное динамическое программмирование

Воспользуемся динамическим программированием: пусть $dp_{l,r,wh}$ — минимальный объём потерянной нефти, пока будут чиниться пробоины с l по r, и после этого робот будет находится в позиции $wh = l$ или $wh = r$.

Можно сделать переход в $dp_{l-1,r,l-1}$, добавив к ответу $|a[wh] - a[l-1]| \cdot (\sum_{i=1}^{n}(d_i) - \sum_{i=l}^{r}(d_i))$, или перейти в $dp_{l,r+1,r+1}$, добавив к ответу $|a[wh] - a[r+1]| \cdot (\sum_{i=1}^{n}(d_i) - \sum_{i=l}^{r}(d_i))$.

Итоговый ответ будет равен $dp_{1,n,1}$ или $dp_{1,n,n}$.

Итоговая асимптотика: $O(n^2)$.

Задача J. Борис и наблюдения

Автор задачи: Андрей Мищенко
Разбор: Андрей Мищенко
Теги: сложные структуры данных, метод «разделяй и властвуй»

Используем метод «разделяй и властвуй». Пусть для массивов a и b требуется быстро производить объединение и поиск ответа.

Достаточно поддерживать четыре значения:

- ans — ответ в массиве;
- max_pref — максимальная сумма на префиксе;
- max_suf — максимальная сумма на суффиксе;
- sum — сумма всех чисел.

При наличии указанных значений появляется возможность быстро объединять два массива и вычислять обновлённые четыре значения для нового массива.

Реализуем «разделяй и властвуй» и вычислим эти значения.

При обновлении некоторого элемента будем так же, как при обслуживании дерева отрезков, спускаться до элемента за $O(log(n))$, а во время подъёма будем пересчитывать обновлённые значения.

Итоговая асимптотика: $O(2 \cdot n + q \cdot log(n))$.