Задача A. Seat Allocation

Идея: Андрей Коробко
Разработка: Андрей Коробко
Разбор: Андрей Коробко
Теги: сортировки, жадные алгоритмы

Подзадача 1. Мы можем образовать только одну пару, и ответом будет сумма её элементов.

Ключевая идея. Оптимальным распределением по парам будет на каждом шаге выбирать текущие минимальный и максимальный элементы массива. Поэтому достаточно отсортировать массив, тогда ответ равен $\max\limits_{1 \leqslant i \leqslant \frac{n}{2}} (a_i + a_{n-i+1})$.

Подзадача 2. Для сортировки массива можно использовать любую квадратичную сортировку, например, сортировку выбором.

Асимптотика такого решения: $O(n^2)$.

Подзадача 3. Массив изначально отсортирован, поэтому остаётся лишь вычислить максимальную сумму в паре, как было указано ранее.

Асимптотика такого решения: $O(n)$.

Подзадача 4. Поскольку числа достаточно малы, мы можем воспользоваться сортировкой подсчётом.

Асимптотика такого решения: $O(n + A)$.

Полное решение. Для полного решения можно воспользоваться любой логарифмической сортировкой, например, встроенной в C++ функцией `std::sort`.

Асимптотика такого решения: $O(n \log n)$.

Задача B. Digital Root

Идея: Андрей Коробко
Разработка: Андрей Коробко
Разбор: Андрей Коробко
Теги: математика, исследование

Подзадача 1. Ответом является цифровой корень числа l.

Подзадача 2. Поскольку все числа состоят из одной цифры, ответом будет сумма чисел от l до r.

Подзадача 3. Заметим, что цифровой корень числа, полученного конкатенацией чисел от l до r, совпадает с цифровым корнем числа, равного сумме чисел от l до r. Поэтому пройдём по заданному отрезку, вычислим сумму чисел и выведем цифровой корень полученной суммы.

Подзадача 4. Как и в предыдущей подзадаче, вычислим сумму чисел от l до r и найдём цифровой корень полученной суммы. Заметим, что при этом необязательно перебирать все элементы, а сумму на отрезке $[l, r]$ можно получить по формуле суммы членов арифметической прогрессии: $\frac{r(r+1)}{2} - \frac{l(l-1)}{2}$. Однако значение суммы от 1 до 10^{18} превосходит вместимость 64-битных целых, что является проблемой в некоторых языках программирования. Поэтому рассмотрим более оптимальное решение.

Полное решение. Исследовав значения цифровых корней натуральных чисел, можно заметить, что эта последовательность имеет вид: `1 2 3 ... 8 9 1 2 3 ...`

Следовательно, мы можем легко посчитать сумму цифровых корней на отрезке $[l, r]$. Нетрудно догадаться, что цифровой корень этой суммы совпадает с числом, полученным конкатенацией чисел от l до r.

Пусть $f(l)$ — сумма цифровых корней для всех чисел от 1 до l. Тогда

$$f(l) = \lfloor \frac{l}{9} \rfloor \cdot 45 + (l \bmod 9) \cdot \frac{(l \bmod 9) + 1}{2}$$

Отсюда сумма цифровых корней на отрезке $[l, r]$ равна $f(r) - f(l-1)$. Ответом будет цифровой корень полученной разности.

Задача C. Connected Network

Идея: Иван Гладченко
Разработка: Андрей Коробко
Разбор: Андрей Коробко
Теги: графы, компоненты связности

Подзадача 1. Переберём все возможные пары вершин и все возможные пути между ними, например, перебором всех перестановок. Если между парой вершин существует хотя бы один путь с ценностью не ниже k, то прибавим её к ответу.

Подзадача 2. Посчитаем число рёбер с весом не менее k, обозначим его как x. Тогда ответом будет $\frac{x(x+1)}{2}$.

Асимптотика такого решения: $O(m)$.

Подзадача 3. Представим наше дерево как массив, предпросчитаем для каждой вершины первую справа вершину, из которой исходит ребро со стоимостью менее k, и обозначим это значение как $next_i$. Ответом на задачу будет $\sum\limits_{i=1}^{n} \frac{(next_i - i)\cdot(next_i - i + 1)}{2}$.

Асимптотика такого решения: $O(n)$.

Полное решение. Давайте изначально не будем добавлять ребра с весом менее k, поскольку они не могут участвовать ни в одном пути требуемой стоимости. Тогда наш граф распадётся на компоненты связности, внутри каждой из которых для любых двух вершин существует путь ценностью не менее k.

Посчитаем размеры этих компонент. Пусть размер текущей компоненты равен c, тогда число пар вершин в ней, для которых существует путь ценностью не менее k, равно $\frac{c(c-1)}{2}$. Просуммировав это значение по всем компонентам, получим ответ на задачу.

Асимптотика такого решения: $O(n)$.

Задача D. Strategy Choice

Идея: Андрей Коробко
Разработка: Андрей Коробко
Разбор: Андрей Коробко
Теги: комбинаторика, числа Каталана, динамическое программирование

Подзадача 1. Можем вручную посчитать ответы для всех значений n и сохранить их в виде констант.

Подзадача 2. Заметим, что бинарные строки можно представить в виде битовой маски. Переберём все возможные битовые маски длины n и для каждой проверим, выполняется ли условие задачи, если выполняется — увеличим ответ на одну последовательность, иначе перейдём к следующей.

Асимптотика такого решения: $O(2^n \cdot n)$.

Подзадача 3. Воспользуемся следующей динамикой, пусть $dp_{i,j,k}$ — число подходящих последовательностей длины i, в которых первый игрок выиграл j раундов, а второй — k (при $j < k$ считаем, что $dp_{i,j,k} = 0$). Базовым значением динамики будет: $dp_{0,0,0} = 1$, а ответом $\sum\limits_{i+j \leqslant n} dp_{n,i,j}$.

В такой динамике мы можем перейти из состояния $(i-1, j-1, k)$, добавив победу первому игроку, или из $(i-1, j, k-1)$, добавив победу второму игроку, то есть $dp_{i,j,k} = dp_{i-1,j-1,k} + dp_{i-1,j,k-1}$.

Асимптотика такого решения: $O(n^3)$.

Подзадача 4. Напишем динамику $dp_{i,j}$ — число подходящих последовательностей длины i, в которых разница побед первого и второго игрока равна j. В качестве базы динамики будем считать $dp_{0,0} = 1$, а ответом будет $\sum\limits_{i=0}^{n} dp_{n,i}$.

При переходе мы либо даём победу первому игроку, либо второму игроку, поэтому $dp_{i,j} = dp_{i-1,j-1} + dp_{i-1,j+1}$.

Асимптотика такого решения: $O(n^2)$.

Полное решение. Давайте рассмотрим динамику из предыдущей подзадачи. Заметим, что при чётных i ненулевые значения $dp_{i,j}$ лежат только на чётных j, а при нечётных i — только на нечётных j.

Отсюда следует, что при переходе с нечётного i на чётное каждое значение динамики добавляется к двум состояниям нового слоя, и поэтому $ans_i = 2ans_{i-1}$ для чётного i.

В случае нечётного i аналогично каждому значению динамики дважды вносится вклад в новый слой, за исключением $dp_{i-1,0}$ (переход из состояния $(i-1, 0)$ в $(i, -1)$ мы не учитываем). Поэтому $ans_i = 2ans_{i-1} - dp_{i-1,0}$ для нечётного i.

Рассмотрим теперь $dp_{i,0}$ для всех чётных i, то есть число последовательностей длины i из $\{-1, 1\}$, у которых нет префикса с отрицательной суммой, а общая сумма равна нулю. Это эквивалентно числу правильных скобочных последовательностей из $\frac{i}{2}$ пар скобок, которое равно $\frac{i}{2}$-му числу Каталана.

Следовательно, для нечётного i переход можно записать как $ans_i = 2ans_{i-1} - C_{\frac{i}{2}}$, где C_k — k-е число Каталана.

Ответ на задачу — значение ans_n. Предпросчитав факториалы и их обратные (для чисел Каталана), мы получаем решение за $O(n)$.

Задача E. Array Rebuild

Идея: Андрей Коробко
Разработка: Андрей Коробко
Разбор: Андрей Коробко
Теги: решето Эратосфена, предпросчёт

Подзадача 1. При каждом запросе будем явно вычислять сумму простых делителей числа $\frac{a_1}{k_i}$ — это и будет ответом. Если же a_1 не делится на k_i, ответом будет c.

Подзадача 2. Предпросчитаем количество элементов массива, равных степеням двойки: пусть cnt_i — число элементов a, равных 2^i. Если число из запроса не является степенью двойки, ответ равен $n \cdot c$. Иначе пусть запрошенное число равно 2^j, тогда ответ вычисляется как $\sum\limits_{i=0}^{j-1} cnt_i \cdot c + \sum\limits_{i=j}^{20} cnt_i \cdot (i-j)$.

Асимптотика такого решения: $O(n + q \log A)$.

Подзадача 3. Каждый запрос, будем явно проходиться по массиву, и если a_i и значение из запроса взаимнопростые, то прибавим к ответу c, иначе прибавим к ответу сумму простых делителей числа $\frac{a_i}{k}$.

Асимптотика такого решения: $O(q \cdot n\sqrt{A})$.

Подзадача 4. Для всех чисел менее 1 000 предпросчитаем сумму их простых делителей. Заведём массив ans — минимальная стоимость, чтобы привести все числа массива к x. Тогда, если a_i нельзя привести к x, то мы прибавляем к ans_x значение c, иначе — сумму простых делителей числа $\frac{a_i}{x}$. Все x будем перебирать явно. Для ответа на запрос будем использовать массив ans.

Асимптотика такого решения: $O(nA + q)$.

Полное решение. С помощью решета Эратосфена предпросчитаем для всех чисел до 10^6 сумму их простых делителей. Затем пройдёмся по массиву и для каждого элемента разложим его на делители. Рассмотрим каждый делитель d числа a_i. В таком случае a_i может стать равным $\frac{a_i}{d}$, и мы прибавим к $ans_{\frac{a_i}{d}}$ сумму простых делителей числа d, где ans_j — минимальная стоимость приведения всех приводимых элементов массива к значению j.

Также будем поддерживать массив cnt_j — число элементов, которые можно привести к j. Тогда ответом на запрос будет $ans_k + (n - cnt_k) \cdot c$.

Асимптотика такого решения: $O(n\sqrt{A} + q)$.

Задача F. WB Tree

Идея: Андрей Коробко
Разработка: Андрей Коробко
Разбор: Андрей Коробко
Теги: динамическое программирование на дереве

Подзадача 1. Переберём все подмножества вершин, проверим, являются ли они связными подграфами, и посчитаем, сколько из выбранных вершин чёрные. Сохраним результаты в массиве $cnt_{i,j}$ — количество подграфов из i вершин, из которых j чёрные. Тогда для ответа на запрос достаточно проверить, что $cnt_{k_i,x_i} > 0$.

Асимптотика такого решения: $O(2^n \cdot n + q)$.

Подзадача 2. Воспользуемся динамикой $dp_{v,i,j}$ — существует ли связный подграф, содержащий вершину v, состоящий из i вершин, из которых j чёрные. Будем вычислять её при обходе дерева, для листа $dp_{v,1,0} = 1$ или $dp_{v,1,1} = 1$ в зависимости от цвета вершины. Чтобы посчитать динамику для не листовой вершины v, сначала вычисляем значения динамики для её детей, а затем наивно объединяем их, после чего наивно объединим значения из сыновей вершины за сумму квадратов размеров их поддеревьев.

Асимптотика такого решения: $O(n^5 + q)$.

Подзадача 3. Оптимизируем слияние вершин при помощи рюкзака на дереве, тем самым суммарное объединение всех вершин будет занимать порядка $O(n^3)$.

Асимптотика такого решения: $O(n^3 + q)$.

Полное решение. Заметим, что в динамике из предыдущей подзадачи при фиксированных v и j значения i, для которых существует связный подграф, образуют один непрерывный отрезок. Следовательно, можно вести такую динамику: $dp_{v,i,0}$ — минимальный размер связного подграфа, содержащего вершину v и ровно i чёрных вершин, и $dp_{v,i,1}$ — максимальный размер такого подграфа соответственно.

Рассмотрим, почему такое решение укладывается в ограничения, будем делать это на примере бинарного дерева, так как это легко обобщается на произвольное дерево. Пусть мы хотим объединить два поддерева в вершинах a и b, тогда мы затратим на это в худшем случае $sz_a \cdot sz_b$, где sz_v — размер поддерева вершины v. Оценим суммарное значение этого выражения по всем вершинам. Заметим, что поддеревья двух детей одной вершины не пересекаются. Пусть $sz_a = k$, тогда $sz_b = n - k - 1$, максимум произведения $k(n-k-1)$ достигается примерно при $k \approx \frac{n-1}{2}$ и равно $\frac{(n-1)^2}{4}$. Если дерево сбалансировано, что является худшим случаем, то на глубине d у нас примерно 2^d вершин, а их суммарный вклад в асимптотику — $\frac{n^2}{2^{d+2}}$. Суммарно по всем уровням это $n^2 \cdot \sum\limits_{d=0}^{\log n} \frac{1}{2^{d+2}} \leqslant \frac{n^2}{2}$. Поэтому, пересчёт динамики по всем вершинам не превосходит $O(n^2)$.

После того как мы вычислили динамику, чтобы быстро отвечать на запросы, для каждого количества чёрных вершин переберём все вершины и будем поддерживать границы отрезка размеров подграфов, в которых данное фиксированное количество чёрных вершин встречается. Тогда для ответа на запрос достаточно проверить, что для k_i из запроса оно входит в соответствующий отрезок размеров подграфов, состоящих из x_i чёрных вершин.

Асимптотика такого решения: $O(n^2 + q)$.