Разбор дистанционного тура 2

Yandex Algo, C, 2023

2 декабря 2023 г.

A Оптом — дешевле!

Для того, чтобы решить задачу на 60 баллов, достаточно было перебрать количество покупаемых пакетов трафика на A мегабайт i и на C мегабайт j за $O(N^2)$. Если $A \cdot i + C \cdot j \geq N$, то мы уже набрали достаточное количество трафика, и итоговая цена такого набора $B \cdot i + D \cdot j$, иначе к ней необходимо ещё добавить пакеты трафика по 1 мегабайту в количестве $N - (A \cdot i + C \cdot j)$. Среди всех итоговых цен нужно взять минимальную — это и будет ответом на задачу.

Для полного решения необходимо заметить, что можно перебирать лишь одну переменную, например, i. Когда мы зафиксировали количество пакетов трафика по A мегабайт, все оставшиеся мегабайты (кроме, быть может, некоторого остатка $< C$) нам выгодно заполнить пакетами по C мегабайт, поскольку нам гарантируется, что $C < D$, а значит, цена 1 мегабайта в пакете всегда меньше одного рубля. То есть кандидатов на наиболее выгодный вариант всего два: это либо набрать пакетов по C мегабайт, немного недобрав до N, а остаток «добить» пакетами по 1 мегабайту; либо взять на один больше пакет с C мегабайт, получив $> N$ мегабайт. Поэтому для фиксированного i оптимальная цена:

$$B \cdot i + \left[\frac{N - A \cdot i}{C}\right] \cdot D + min((N - A \cdot i) \mod C, D)$$

Таким образом, получили решение за $O(N)$.

B Задача из ЕГЭ

Обозначим исходное число K. Разберём три возможных случая:

1) N двузначное, т.е. $N = \overline{ab}$. Тогда сначала необходимо проверить, что $a \geq b$ (иначе числа не были записаны в порядке убывания, и ответ на задачу 0). Далее, если $b = 0$, то такая сумма могла получиться только во второй половине K, так как в K нет ведущих нулей. Тогда оптимальное $K = \overline{1(a-1)00}$. Иначе оптимальное $K = \overline{1(b-1)0a}$.

2) N трёхзначное, т.е. $N = \overline{abc}$. Тогда, поскольку части числа записаны по убыванию, одна из сумм цифр в K $\overline{ab}$, а вторая c. Заметим, что сумма двух цифр не может быть > 18, поэтому если $\overline{ab} > 18$, то необходимо сразу вывести 0. Если же $\overline{ab} \leq 18$, то проверяем, равно ли c 0, и если да, оптимальное $K = \overline{(\overline{ab} - 9)900}$, иначе оптимальное $K = \overline{1(c-1)(\overline{ab} - 9)9}$.

3) N четырёхзначное, т.е. $N = \overline{abcd}$. Тогда одна из сумм цифр равна $\overline{ab}$, другая равна $\overline{cd}$. Проверим, что $\overline{ab} \geq \overline{cd}; \overline{ab}, \overline{cd} < 18; c \neq 0$ (последнее условие, поскольку мы записываем суммы без ведущих нулей), если что-то из этого не выполняется, выведем 0. Иначе оптимальное $K = \overline{(\overline{cd} - 9)9(\overline{ab} - 9)9}$.

C Тимофей и Бумажки

Наиболее выгодной в этой задаче будет жадная стратегия: сначала будем перекладывать в новую стопку все листы из самой высокой стопки, пока её высота не сравняется со второй по высоте; затем будем перекладывать листы одновременно из двух самых высоких стопок в новую, пока их высота не сравняется с третьей по высоте, и так далее; так будем действовать до момента, пока высота новой стопки не станет $\geq$ высоте текущей самой высокой стопки.
Реализовать данное решение можно за $O(n \log n + \log(\max x_i))$: сначала отсортируем стопки по убыванию, затем будем двигаться по ним слева направо, поддерживая последний номер текущей максимальной стопки i в 1-нумерации (удобно добавить фиктивную стопку с высотой 0 в конец массива). Также будем поддерживать текущую высоту новой стопки h. На каждом шаге алгоритма, если $h + (x_i - x_{i+1}) \cdot i < x_{i+1}$, то сделаем $h+ = (x_i - x_{i+1}) \cdot i$ и подвинем i до последнего элемента, равного x_{i+1}. Иначе бинпоиском найдём минимальное y, такое что $h + y \geq x_i - [y/i]$, тогда ответом на задачу будет $h + y$.

D Баскетбол

Поскольку в этой задаче все числа небольшие, достаточно было аккуратно промоделировать процесс замены игроков. Это можно было реализовать, например, за $O(nm)$ следующим образом: для каждого игрока храним количество минут, проведённых на поле, t_i и булеву переменную on_i, показывающую, на поле сейчас i-й игрок или в запасе. На каждом из m шагов алгоритма достаточно за $O(n)$ прибавить 1 к t_i у всех игроков на поле и сменить значение on_i на противоположное у 4-х игроков по правилам, указанным в условии. Ответ на задачу — фамилии в лексикографическом порядке всех игроков, у которых после m шагов алгоритма перменная on_i равна 1.

E Робот

Отсортируем работы по убыванию дедлайна. Будем двигаться от дня с номером d_1 к дню с номером 1, и каждый раз ставить наиболее подходящего кандидата из ещё нераспределённых задач на текущий день. Поскольку мы хотим минимизировать итоговый штраф, то таким кандидатом на день i будет ещё нераспределённая задача с максимальным штрафом и дедлайном $\geq i$ (если такая существует). Когда мы доходим до дня, в который есть дедлайн у одной или нескольких задач, мы добавляем эти задачи в множество нераспределённых. Чтобы оптимально поддерживать текущее множество нераспределённых задач и доставать из них максимальную по штрафу, можно воспользоваться структурой данных set или priority_queue. Тогда ответ — сумма штрафов всех нераспределённых после дня 1 задач. Получили решение за $O((n + \max d_i) \log n)$.