Задача A. Мобайл

Можно заметить, что если $D \leqslant B$, то ответом является A — ведь это минимальная стоимость по тарифу. Иначе за оставшиеся $D - B$ мегабайт надо доплатить из рассчета C рублей за мегабайт. В таком случае ответ равен $A + (D - B) \cdot C$.

Задача B. Рулет

Пусть $F(n)$ — число разрезов для числа n. $F(1) = 0$.

Заметим, что $F(2^k) = k$, потому что можно k раз совмещать все рулеты и резать их пополам. Также заметим, что $F(n) \leqslant F(n+1)$, потому что разрезание рулета на n частей —подзадача от разрезания рулета на $n+1$ часть. Кроме того, заметим, что за k разрезов можно получить не больше, чем 2^k кусков.

Совместив полученные знания, получаем, что для $n \in (2^{k-1}; 2^k]$ ответом будет число k. Значит, нужно найти старший бит числа n, что можно сделать с помощью цикла, в котором n будет итеративно делиться на два. Число итераций и будет ответом.

Задача C. Лифты и переговорки

Найдем максимум и минимум во входных данных.

Мы знаем, что ответ не меньше чем $max - min$. Если нельзя успеть пройти всех сотрудников за это время, то это значит, что сотрудник успеет уйти. Тогда выгоднее всего сразу доехать до его этажа, а потом сделать одну из двух операций — либо подняться на самый верх, а потом спуститься вниз, либо спуститься на самый низ, а потом подняться наверх. Если уезжающий сотрудник был на этаже x, то получившимися формулами будет $max - min + max - x$ и $max - min + x - min$ соответственно, осталось только найти минимум из этих двух величин.

Задача D. Бумажка с числами

Можно заметить, что задача разбивается на k независимых задач «какую цифру в наборе заменить, чтобы сумма выросла как можно больше?».

Когда мы меняем цифру в m-м разряде с a на b, мы прибавляем к ответу $10^m \cdot (b - a)$. Можно заметить, что в любом числе выгодно менять цифру только на 9. Значит, для каждой цифры в разложении числа на цифры мы знаем, насколько нам выгодно эту цифру менять. Осталось только отсортировать эти операции в порядке убывания выгоды и посчитать выгоду первых k из них.

Задача E. Тестирование

На самом деле, интересных чисел достаточно мало. Можно их все сгенерировать заранее, а потом просто для каждого элемента в списке проверить, что он принадлежит отрезку.

Чтобы сгенерировать весь список, надо для каждой цифры k от 1 до 9 сделать генерацию всех чисел, состоящих только из k. Это делается пошагово: если предыдущее значение было равно x, то следующее — $10 \cdot x + k$.

Задача F. Физкультуа

В задаче требовалось найти все позиции учеников, которые имеют четный рост и стоят на нечетном месте, а также найти все позиции учеников, которые имеют нечетный рост и стоят на четном месте. Все эти ученики стояли не на своих местах. Ребята с четным ростом хотят попасть на нечетную позицию, а ребята с нечетным — на четную. Значит, если размеры обоих списков равны единице, то искомая замена возможна. Иначе — ответ «Нет».

Задача G. Тайный Санта

Если формализовать условие задачи, то можно увидеть, что от нас требуется сделать функциональный граф циклом перенаправлением одного ребра. Функциональный граф — такой ориентированный граф, что из каждой вершины выходит ровно одно ребро.

Посчитаем для каждой вершины, сколько в нее входит ребер. Если есть ровно один ученик, к которому приходят 2 ребра, и другой ученик, к которому приходит 0 ребер, то можно попытаться перенаправить одно из ребер в 0-ученика. Это можно сделать двумя разными способами, и если после одного из вариантов граф оказывается циклом, то эта замена является ответом. Остальные вершины обязательно должны иметь степень 1.

Можно заметить, что больше никаких вариантов перенаправить ребра нет. Это верно, потому что степень ровно одной вершины уменьшится на 1, и степень ровно одной вершины увеличится на 1, а в итоге все степени должны оказаться равны 1.

Для того, чтобы проверить, что после замены граф оказался циклом, достаточно проверить цепочку передачи подарков — она должна быть цикличной и состоять ровно из n элементов.

Задача H. Переговорка

В этой задаче существует формульное решение, которое требует некоторого количества математических знаний и внимательности в работе с формулами. Для начала с помощью обратных тригонометрических величин найдем угол поворота прямоугольника ϕ. Далее выразим глобальные координаты искомой точки (X, Y) через угол, коэффициент подобия прямоугольников k и координатой точки с относительными координатами $(0, 0)$ для второго прямоугольника (x, y):

$$X = x + kX\cos\phi + kY\sin\phi$$

$$Y = y + kY\sin\phi - kX\cos\phi$$

$$X = \frac{x + kY\sin\phi}{1 - \cos\phi}$$

$$Y = \frac{y - k\frac{x+kY\sin\phi}{1-\cos\phi}\cos\phi}{1 - k\sin\phi}$$

Задача I. Фудкорт

Воспользуемся методом динамического программирования: $dp[n][k]$— стоимость, которую потратит Даня за n обедов, если у него осталось k талонов. Тогда формула пересчета имеет вид:

$$dp[n+1][k] = \min(dp[n][k - (if a_n > 100 then 1 else 0)] + a_n, dp[n-1][k+1])$$

Для восстановления ответа кроме минимума требуется хранить тот переход, на котором минимум был достигнут.

Задача J. Торт

Сделаем бинарный поиск по ответу. Если текущий x находится слишком справа, то двигаем его влево, а иначе двигаем вправо.

Чтобы понять, в каких пропорциях сделается разрез, требуется пересечь прямую со сторонами многоугольника и посчитать площади левой и правой части.

Задача K. Математическая задача

Пусть $f(k) = \prod_{x=l}^{k} x \pmod p$, $F(k) = \prod_{x=l}^{k} \frac{1}{x} \pmod p$.

Посчитать $f(k)$ можно за линейное время: $f(k) = f(k-1) \cdot k \pmod p$.

Посчитать $F(r)$ можно с помощью бинарного возведения в степень: $F(r) = (f(r))^{p-2}$. Бинарное возведение в степень считается по правилу: $a^{2k} = (a^k)^2$, $a^{2k+1} = a \cdot a^{2k}$. Такая цепочка в рекурсивной записи требует всего лишь $O(\log k)$ действий для возведения в степень.

Далее обратным ходом вычисляем $F(k) = F(k+1) \cdot k \pmod p$.

Дальше, $\frac{1}{k} = f(k) \cdot F(k-1)$. Осталось только просуммировать эти величины по модулю p.

Задача L. Три монеты

Заранее отсортируем веса монет так, чтобы выполнялось условие $A \leqslant B \leqslant C$. Пусть $d[x]$ — минимальное k такое, что монетами можно набрать вес $kC + x$. Заранее известно, что $dp[1] = 0$ (исходно примем остальные значения за $+\infty$. Для остальных значений $dp[x]$ верны следующие правила:

$$\begin{cases} dp[x] = \min(dp[x], dp[x - A]) & , x \geqslant A \\ dp[x] = \min(dp[x], dp[x - A + C] + 1) & , x < A \end{cases}$$

$$\begin{cases} dp[x] = \min(dp[x], dp[x - B]) & , x \geqslant B \\ dp[x] = \min(dp[x], dp[x - B + C] + 1) & , x < B \end{cases}$$

Заметим, что этот набор правил может только уменьшать значение $dp[x]$, причем применив эти операции достаточно много раз, мы получим оптимальные значения $dp[x]$. Вопрос только в том, в каком порядке применять эти правила.

Давайте посмотрим на остатки, как на вершины графа, а на переходы в правилах — как на ребра веса $0/1$. Тогда $d[x]$ — это просто определение расстояние в получившемся графе, которое можно посчитать алгоритмом Дейкстры.

Имея $d[x]$, посчитать ответ на задачу совсем несложно. Поскольку можно «добить» любую сумму C-шками, то ответом будет $\sum_{x=0}^{C-1} \lfloor \frac{N-x}{C} \rfloor - dp[x]$