Задача A. Набор текста

В задаче нужно аккуратно промоделировать процесс набора текста. Для начала заметим, что набор символа требует нажатия как минимум одной клавиши, но иногда еще нужно нажать клавишу shift. Таким образом ответ будет равен сумме длины строки и числа нажатий клавиши shift.

Для удобства подсчета нажатий можно поддерживать дополнительную логическую переменную, которая хранит информацию нажат shift в данный момент или нет. Тогда существует три класса символов:

- Строчные буквы латинского алфавита, «.», «,» — shift не должен быть нажат. Заметим, что отпустить клавишу не требует нажатия клавиши. Поэтому «бесплатно» инвертируем значение соответвующей логической переменной, если нужно.
- Заглавные буквы латинского алфавита, «!», «?», «$», «(» и «)» — shift должен быть нажат. Если в данный момент shift не нажат, добавляем единицу к ответу, а затем инвертируем значение соответвующей логической переменной.
- Символ пробела — не делаем ничего.

Остается пройти по строке и обработать класс каждого символа.

Задача B. Крупная закупка

Сформулируем задачу кратко: дано n видов оружия, требуется купить m штук, из которых найдется хотя бы k разных видов. При этом требуется максимизировать суммарную мощность оружия, а при равной мощности минимизировать максимальное количество оружия одного вида.

Первым шагом отсортируем виды оружия по убыванию мощности. Пусть теперь $a_1 \geqslant a_2 \geqslant \ldots \geqslant a_n$. Заметим, что имеют место следующие факты:

1. в каком-то оптимальном наборе есть хотя бы по одному экземпляру оружия каждого из первых k видов (потому что всего есть хотя бы k различных видов, и не уменьшая суммарную мощность можно заменить все оружие мощности a_i на оружие мощности $a_{j<i}$)
2. если $a_1 = a_2 = \ldots = a_i \neq a_{i+1}$, то больше одного экземпляра можно брать только у видов $1 \ldots i$, потому что в ином случае суммарная мощность будет не максимальна
3. таким образом, если $i \leqslant k$, следует взять по одному экземпляру каждого из первых k видов, а оставшиеся $m - k$ наиболее равномерно распределить по первым i видам
4. если же $i > k$, то выгодно наиболее равномерно распределить все m штук по первым i видам, так как это минимизирует максимальное количество одного и того же вида

Подгруппа *(1)* решалась первыми двумя наблюдениями, достаточно было понять, что надо взять по одному экземпляру каждого из k наиболее мощных видов, и еще $m - k$ экземпляров самого мощного.

Подгруппа *(2)* требовала выбрать по одному экземпляру каждого из k максимальных видов, а в подруппе *(3)* следовало заметить, что если первые $i > k$ видов одинаковы по мощности, в оптимальном ответе будет i видов, а не k.

Полное решение заключается в проверке отношения между k и количеством максимальных по мощности видов оружия i, и выводе $\max(i, k)$ первых видов и распределения оставшихся $m - \max(i, k)$ экземпляров примерно поровну (с отличием количества разных видов не более, чем на один) по i абсолютно максимальным по мощности видам оружия.

Задача C. Поместье мафии

Задачу можно решить методом двоичного поиска по ответу. Научимся проверять, можно ли разбить отрезок на n частей так, чтобы длины всех частей не превосходили x. Тогда ответом будет минимальное x, при котором будет существовать разбиение.

Чтобы проверить существование разбиения при выбранном x, нужно действовать жадно. Начало самого левого отрезка имеет координату 0, а концом сделаем точку с координатой $\min(a_2, x)$. Заметим, что конец точно не может иметь большую координату. Потому что тогда либо точки a_1 и a_2 обе будут лежать целиком внутри первого отрезка, либо первый отрезок будет иметь длину больше x. Также, несложно заметить, что уменьшать координату конца не имеет смысла, потому что если существует разбиение при котором конец первого отрезка имеет меньшую координату, эту координату можно увеличить. При этом, длина первого отрезка не станет больше x, длина второго отрезка уменьшится, длины остальных отрезков не изменятся. Поэтому это разбиение тоже будет корректно. Таким образом, можно продолжать жадный алгоритм и для следующих отрезок, рассматривая в качестве координаты начала конец предыдущего отрезка. Если какой-то отрезок не содержит ни одной точки, либо конец последнего отрезка имеет координату меньше, чем l, разбиения при данном x не существует.

Задача D. Счастливые билетики

Рассмотрим бинарную операцию op, которая будет вычислена в последнюю очередь. Тогда все выражение будет равно $op(expr_1, expr_2)$, где $expr_1$ — значение выражения слева от операнда, а $expr_2$ — справа. Таким образом, для каждой строки из цифр s можно найти множество значений $vals[s]$, которые могут получиться, если расставить в этой строке операции и скобки. Это может быть либо просто число s, либо $op(a, b)$, где op — любая операция, $a \in vals[s_1]$, $b \in vals[s_2]$, $s = s_1 s_2$.

Несложно заметить, что $|vals[s]| \leqslant 2 \cdot 10^{len(s)}$, потому что для любой операции $len(op(a, b)) \leqslant len(a) + len(b)$, где $len(x)$ — количество десятичных цифр в числе x.

Также, заметим, что для строк длины 6 не нужно искать всё множество $vals$, а достаточно только проверить, принадлежит ли ему 100. Аналогично, для множеств для строк длины 5 не нужно находить множество целиком, достаточно найти только значения, которые при применении операции с числом длины 1 могут дать 100.

Также, нужно не забыть про операцию отрицания.

Такое решение укладывается в ограничения и находит все решения.