

Задача А. Йода и приставки Си

Данная задача разбивается на две части:

- Нужно распарсить последовательность приставок в префиксе, а также размерность величины. Можно по очереди сравнивать по четыре буквы префикса со списком четырех-буквенных приставок и, если такой последовательности из четырех букв не будет, то попробовать взять пять. Если встретили последовательность «meter», проверяем последний символ на равенство «2» или «3» и таким образом узнаем размерность.
- Считаем сумму чисел соответствующих приставкам чисел и умножаем на размерность, если нужно.

Задача В. Джедайские вычисления

Для решения первых групп можно было явно перебрать строки с одним удаленным символом и воспользоваться функцией `eval` в Python. Могли быть проблемы с ведущими нулями, появляющимися после удаления первой цифры в каком-то из чисел, которые можно было решить с помощью регулярного выражения, которое мы здесь приводить не будем.

Для полного решения можно было заметить, что удаление одного символа затрагивает в нем не более двух чисел, а значит можно не пересчитывать все выражение заново:

- при удалении цифры меняется только то число, из которого цифра была удалена
- при удалении знака '+' или '-' два числа вокруг заменяются на их конкатенацию

Сделаем следующее: посчитаем значение исходного выражения, просто пройдясь по составляющим его числам с учетом стоящих перед ними знаков, а затем переберем удаляемый символ и найдем максимум разницы между значением выражения после удаления и исходным значением.

Для того, чтобы за $\mathcal{O}(1)$ находить разницу, будем хранить числа в строковом виде. Тогда, используя перевод строки в число с помощью простого цикла, можно будет просто находить значения конкатенаций двух чисел и чисел с удаленной цифрой:

- чтобы посчитать разницу при удалении цифры, достаточно из нового числа вычесть старое и взять с нужным знаком
- чтобы посчитать разницу при удалении знака между двумя числами, надо взять разницу с нужными знаками нового числа (полученного конкатенацией) и двух исходных чисел

Ответом будет сумма исходного значения выражения и максимальной достижимой удалением одного символа разницы.

Задача С. Древний замок

Заметим, что каждый камень можно коснуться всего с 4 сторон. Тогда посчитаем $dp_{i,j}$ — минимальное расстояние, что мы коснулись первых i камней, и коснулись последнего со стороны j . Для того, чтобы перейти от $dp_{i,j}$ к $dp_{i+1,q}$ надо посчитать расстояние от одной клетки до другой при помощи алгоритма BFS за время $\mathcal{O}(n \cdot m)$. Отдельно надо посчитать расстояние от старта до первого камня и от последнего камня до финиша. Значит у нас получилось $\mathcal{O}(k)$ запусков BFS. Значит итоговое время работы $\mathcal{O}(n \cdot m \cdot k)$.