Вы получаете шаблон как положительное целое число, и вам также дается строковая структура как слово. Для сравнения система распознавания должна преобразовать целочисленный шаблон в двоичную форму. Нужно добавить нули слева, чтобы соответствовать длине структуры, и сравнить эту комбинацию со структурой.

1 - это буква, а 0 - цифра.

Если шаблон и структура совпадают, верните значение True, иначе верните значение False. Если шаблон больше заданной структуры, то они не совпадают (пример: 8 = 1000 и "a").

Вот пример: данный шаблон 42 и структура "12a0b3e4". 42 == 101010 в двоичной форме, но этого недостаточно, чтобы соответствовать структуре. Добавляем нули влево, чтобы получить "00101010". Теперь сравните два:

     42 == 00101010  
12a0b3e4 == DDLDLDLD  
--------------------  
   True    VVVVVVVV

Вот еще один пример - 101 и «ab23b4zz»:

   101 == 01100101  
ab23b4zz == LLDDLDLL  
--------------------  
  False    XVXVXXXV

Предпосылка:

Структура сопоставляется выражением "\ A [A-Za-z0-9] {1, 32} \ Z" regexp.

Pattern\_level = 2

Ранг 2:

На уровне шаблона по умолчанию (level = 2) мы используем двоичную форму. Если уровень patter равен 3, затем преобразуйте целочисленный шаблон в тройную систему счисления (base 3). 0 - это цифра, 1 - строчная буква, 2 - прописная буква.

Предпосылка:

Структура сопоставляется выражением "\A[a-zA-Z0-9]{1, 32}\Z" regexp

pattern\_level ∈ {2, 3}

Ранг 3: Если уровень паттерна равен 4, то преобразование целочисленного паттерна в четвертичную систему счисления (база 4). 0 - это цифра, 1 - строчная буква, 2 - прописная буква, 3 - пробел.

Предпосылка:

Структура сопоставляется выражением "\A[ a-zA-Z0-9]{1, 32}\Z" regexp.

pattern\_level ∈ {2, 3, 4}

Пример:

checkStructure(42, "12a0b3e4") === true  
checkStructure(101, "ab23b4zz") === false