Задание (вариант 18):

Даны регулярное выражение в обратной польской записи R и буква x. Найти максимальное k, такое что в языке L есть слова, начинающиеся с x^k. Алфавит: $\{a, b, c, 1, ., +, *\}$

Алгоритм: Для каждого регулярного выражения r, с которым работаем во время перевода регулярного выражения из обратной польской записи в обычную запись, будем хранить два параметра:

$$\begin{cases} max\_pref\_len \in [0, inf] & \text{т.е. } max\ k : x^k \text{ – префикс } w \in r \\ max\_word\_len \in [-1, inf] & \text{т.е. } max\ m : x^m = w' \in r \end{cases}$$

В написанном в этом репозитории алгоритме $inf = 10^5$. Очевидно, что всегда верно $max\_pref\_len \geq max\_word\_len$. Поянсим границы для каждого из параметров. Для всех примеров положим $x = a$.
Параметр max_pref_len принимает значения:

1. 0, если r не задает слово, в котором есть префикс, состоящий из буквы x. (Например, $r = ba$)
2. inf, если r задает слово, в котором есть префикс из буквы x, размер которого превышает заранее зафиксированное число. (Например, $r = a^*b$)

Параметр max_word_len принимает значения:

1. -1, если r не задает слово, полностью состоящее из букв x. (Например, $r = ab$)
2. 0, если r не задает слово, полностью состоящее из букв x, но при этом задает пустое слово, то есть 1. (Например, $r = ab + 1$)
3. inf, если r задает слово, полностью состоящее из букв x, размер которого превышает заранее зафиксированное число. (Например, $r = a^*$)

Тогда база будет вот такой (r – однобуквенное выражение, то есть $r = letter$):

- Если $letter = x$, то
  - $max\_pref\_len = 1$
  - $max\_word\_len = 1$
- Если $letter \neq x$, то
  - $max\_pref\_len = 0$
  - $max\_word\_len = -1$

Теперь разберем, каким образом определяются значения $r_{max\_pref\_len}$ и $r_{max\_word\_len}$ для $r = r_1 \# r_2$, где $\#$ – это одна из операций $\{+^{(2)}, .^{(2)}, *^{(1)}\}$, а r_1, r_2 – корректные регулярные выражения, для которых уже посчитаны $r_{i\,max\_pref\_len}$ и $r_{i\,max\_word\_len}$ $\ i \in [1, 2]$. Будем называть $w \in r$ – слово с максимальным префиксом или максимальное слово, полностью состоящее из букв x.

1. Если это операция $+^{(2)}$, то в терминах регулярных выражений для получения слова w выбирается лучшее слово из r_1 или r_2. Так как параметры независимые, то
   - $r_{max\_pref\_len} = \max(r_{1\,max\_pref\_len},\ r_{2\,max\_pref\_len})$
   - $r_{max\_word\_len} = \max(r_{1\,max\_word\_len},\ r_{2\,max\_word\_len})$
2. Если это операция $.^{(2)}$, то в терминах регулярных выражений для получения слова w склеиваются слова из r_1 и r_2. Тогда, возможны разные случаи:

- Если r_1 задает слово, полностью состоящее из букв $x \Leftrightarrow \mathrm{r}_{1\,max\_word\_len} \neq -1$, то $r_{max\_pref\_len}$ высчитывается как максимум из {склеивания всего слова из букв x из r_1 и префикса из r_2} или {префикса из r_1}, следовательно формула такая: $r_{max\_pref\_len} = \max(r_{1\,max\_pref\_len},\, r_{1\,max\_word\_len} + r_{2\,max\_pref\_len})$
- Иначе, префикс r_2 не имеет смысла рассматривать, так как в r_1 после префикса вида $x^{r_{1\,max\_pref\_len}}$ идут другие буквы, и тогда $r_{max\_pref\_len} = r_{1\,max\_word\_len}$
- Если r_1 и r_2 содержат слова, состоящие только из букв x, то $r_{max\_word\_len}$ высчитывается как сумма длин длинейших слов, состоящих только из букв x, из обеих регулярных выражением, то есть $r_{max\_word\_len} = r_{1\,max\_word\_len} + r_{2\,max\_word\_len}$
- Иначе, слов, полностью состоящих из букв x, появиться не может и параметр $r_{max\_word\_len} = -1$

3. Если это операция $*^{(1)}$, то в терминах регулярных выражений для получения слова w склеиваются слова из r_1 сами с собой необходимое число раз. Пусть для упрощения обозначения $p = r_{1\,max\_word\_len}$. Тогда, возможны случаи:
   - Если r_1 содержит слово вида x^p (без учета пустого слова, то есть $p \neq 0$), то тогда мы можем получить префикс размера inf, слеивая x^p само с собой кучу раз (для $p = 0$, очевидно это неверно). Следовательно, если $r_{1\,max\_word\_len} > 0$, то $r_{max\_pref\_len} = r_{max\_word\_len} = inf$
   - В противном же случае $(p \leq 0)$ $r_{max\_pref\_len} = r_{1\,max\_pref\_len}$, так как оператором $*$ мы никак не будем разрастать слово (это бессмысленно), а $r_{max\_word\_len} = 0$, так как оператор $*$ гарантирует наличие пустого слова в r.

В итоге, обрабатывая с помощью стека исходное регулярное выражение в обратной польской записи, мы придем к R в обычной записи и у нас будут посчитаны два параметра $R_{max\_pref\_len}$ и $R_{max\_word\_len}$. Логично, что ответом на задачу является максимум из этих двух чисел. Однако, как было замечено в начале алгоритма, $max\_pref\_len \geq max\_word\_len$, значит максимум всегда будет возвращать первое число. Таким образом, ответом будет являться $R_{max\_pref\_len}$.

Корректность: Из алгоритма видно, что задача решается индукцией по увеличению приоритета операций над регулярным выражением. Так как исходное регулярное выражение задается нам в обратной польской нотации, то операции поступают к нам сразу в интересующем нас порядке.

- Очевидно, что база индукции (значения параметров для букв) корректна.
- Предполагаем, что параметры были вычислены корректно на всех шагах.
- Переход: делаем последнюю операцию, после выполнения которой, мы получим требуемое регулярное выражение. Так как при парсинге каждой из трех возможных операциий наш алгоритм максимизирует ответ по очевидной логике, то в итоге мы получаем величину максимального префикса состоящего из букв x.

Асимптотика: Пусть $len(R)$ – это длина задаваемого регулярного выражения в обратной польской нотации. Заметим, что обработку выражения мы делаем с помощью стека, в котором операции $push$, top и pop работают за $O(1)$. Мы идем поэлементно по R и взаимодействуем со стеком (кладем в него что-то единожды, либо достаем 1-2 элемента, пересчитываем два параметра за $O(1)$ и кладем это в стек). Значит, один символ из R обробатывается за $O(1)$. Следовательно, итоговая асимптотика $O(len(R))$.