

Сначала нужно построить префикс-функцию для pattern, который будем искать.
На i -й позиции prefix-функции стоит длина наибольшего общего префикса и суффикса строки $s[0...i]$ (строки, заканчивающейся на текущем i -м символе).

$Prefix[i] = \max(S[0...k - 1] == S[i - k + 1...i])$ по k для $0 \leq k < i + 1$

После построения префикс-функции для паттерна:

- 1) Инициализируем два указателя для s и для pattern
- 2) Если текущие символы равны, двигаем оба указателя вперед
- 3) Если достигли конца паттерна (собрали весь), то возвращаем индекс вхождения
- 4) Если текущие символы не равны,
если указатель по паттерну > 0 : передвигаем указатель по паттерну пользуясь префикс-функцией **(Это означает, что только что мы сравнивали паттерн с начала с частью строки и дошли до позиции j в паттерне. Однако, встретили несовпадающий символ. Тогда мы сдвигаем указатель назад по паттерну на $prefix[j - 1]$ на позицию, соответствующую длине наибольшего префикса, равного суффиксу уже совпавшей части. Это позволяет продолжать сравнение, сохраняя уже проверенные символы. Грубо говоря, пусть $s = \text{«ababaka»}$, $pattern = \text{«abak»}$**
Когда мы будем находиться в состоянии **«abab»** и **«abak»** и последние буквы «b» и «k» не совпадут, то перейдем в состояние **«ab»** и **«ab»**. То есть начинаем дальнейшее сравнение уже с существующего суффикса, равному префиксу)
иначе двигаем указатель по строке **(Это значит, что текущая буква в строке не совпадает с первой в паттерне, и просто идем дальше)**

Временная сложность: $O(N + M)$, N - длина строки, M - длина паттерна.

В цикле КМП проходим по строке длины N , смена индексов - $O(1)$.

Предварительное построение префикс-функции - $O(M)$