**Ахо-Корасик.** Хотим решать следующую задачу: Дан набор из n строк  $s_i$  суммарного размера L на алфавите  $\sigma$ . Кроме этого, дан текст t, и задается вопрос насчет вхождений: сколько строк из набора входят в t как подстроки?

Первый шаг: построить бор на наборе строк. Теперь мы хотим сделать на этом боре детерменированный автомат. Автомат при чтении  $t_0 \dots t_i$  должен переводить нас в вершину, которой соответствует самый длинный суффикс строки  $t_0 \dots t_i$  из присутствующих в боре.

Для каждой вершины v создадим суффиксную ссылку link, которая будет вести в вершину, соответствующую самому длинному суффиксу str(v), не совпадающему с str(v), но присутствующему в боре. Кроме этого, обозначим переходы автомата за  $go_c$ . Тогда понятно, что  $link_v$  можно найти, если посмотреть на суффиксную ссылку предка. А именно, если из суффиксной ссылки предка есть переход по символу, который написан на ребре  $(p_v, v)$ , то  $link_v$  ведет именно туда. Если там перехода не было, то надо посмотреть на следующего предка в дереве суфссылок, и сделать аналогичную операцию. Но, если считать, что для предков посчитано  $go_c$ , то это то же самое, что взять  $go_{c(p_v,v)}$  для  $link_{p_v}$ .

Для того, чтобы считать переходы в автомате, тоже можно воспользоваться предыдущими значениями. А именно, если из v есть ребро с символом c, то тогда  $go_c$  будет указывать в конец ребра. Иначе следует посмотреть на  $go_c$  для нашей суффиксной ссылки.

Обе динамики можно посчитать либо лениво, либо с помощью обхода в порядке возрастания глубин.

**Оценка времени работы** Можно оценить как  $O(m|\sigma|)$ , где m — число вершин в дереве, потому что динамика занимает ровно столько памяти (тривиально), и для нее верно time = memory. Также есть оценка O(L), которую можно получить, если показать, что у нас всего спусков бывает не более чем O(L), а подъемов суммарно не больше, чем спусков.