

# Старобарский рэп

Заметим несколько ключевых фактов:

1. Можно гарантировать ответ не меньше 0.5 — взять суффикс длины 1.
2. Не имеет рассматривать большие суффиксы, если у нас уже есть  $d = 1 + \log_2 |s_i|$  различий. Даже если взять всю строку с  $d$  различиями, получим величину рифмы  $\frac{|s_i|}{2^d} \leq 0.5$ .
3. Если рассмотреть два таких суффикса с длинами  $l_1 < l_2$ , что количество различий в них совпадает, то всегда выгоднее взять более длинный.

Из этого следует основная идея решения: сравнивая два слова  $s_i$  и  $s_j$ , имеет смысл рассмотреть только  $1 + \log_2 \min(|s_i|, |s_j|)$  наименьших суффиксов, следующая буква после которых различается. Среди них посчитать наибольшую величину рифмы и вывести.

Можно развернуть все строки, и для каждой строки  $s = c_1 c_2 c_3 \dots c_l$  посчитать полиномиальные хэши для всех префиксов:  $(c_1 + c_2 A + c_3 A^2 + \dots + c_k A^{k-1}) \bmod M$ . Теперь символы отрезаются с начала строк, а не с конца, и можно легко проверять две подстроки с одинаковым количеством отрезанных символов на равенство. Чтобы сравнить префикс длины  $a$  строк  $s_i$  и  $s_j$  с  $c$  отрезанными символами, нужно у обеих строк посчитать разность хэшей  $(a + c)$ -го префикса и  $c$ -го префикса. Если они равны, то и соответствующие подстроки с большой вероятностью равны.

Осталось у двух строк начиная с  $c$ -го символа найти первые  $d$  отличий. Каждое такое отличие ищется бинарным поиском: начиная с позиции сразу после предыдущего отличия, найдём подстроку наибольшей длины, совпадающую у обеих строк. Как было описано ранее, это позволяет нам найти ответ на запрос. Так как мы ограничили  $d$  логарифмом длины, сложность такого решения равна  $O(N + \sum |s_i| + Q \cdot \log^2(\max(|s_i|)))$ .