

## Паша и пароли

Заметим, что можно сразу избавиться от паролей, которые являются подстроками других строк. Для этого переберем всевозможные пары строк, и проверить, находится ли строка  $s_i$  в строке  $s_j$  ( $i \neq j$ ) с помощью любого известного алгоритма (к примеру, алгоритм КМП или хэши). Также нужно быть аккуратными с одинаковыми строками.

Теперь давайте построим ориентированный граф на  $n$  вершинах. Длина ребра  $(i, j)$  будет равна минимальному количеству символов, которые необходимо дописать в конец  $s_i$ , чтобы полученная строка содержала как подстроку строку  $s_j$ . Давайте научимся считать эту величину: так как никакие 2 строки не содержатся друг в друге, найдем максимальный суффикс  $s_i$ , совпадающий с префиксом  $s_j$  такой же длины. Тогда несложно понять, что искомая величина — разность длины  $s_j$  и длины максимального совпадающего суффикса  $s_i$  и префикса  $s_j$ . А максимальные совпадающие префикс и суффикс можно найти, взяв последнее значение префикс-функции от строки « $s_j\#s_i$ », или посчитав аналогичное значение с помощью хэшей.

Теперь заметим, что ответ на задачу — это гамильтонов путь минимальной длины в приведенном выше графе, с начальными степенями: степень  $i$ -й вершины — длина строки  $s_i$ . Это несложно показать: посмотрим на вхождения данных строк в ответ: они идут в каком-то порядке. Очевидно, что нет смысла включать одну строку в этот порядок более 1 раза, ведь на графе выполняется неравенство треугольника. Чтобы для данного порядка минимизировать ответ, необходимо максимизировать «наложение» соседних пар строк — именно это мы и минимизируем в искомом графе.

В итоге мы получаем решение за  $O(2^n * n^2 + S * n^2)$  — поиск гамильтонова пути минимальной стоимости займет  $O(2^n * n^2)$ , и удаление входящих строк изначально —  $O(S * n^2)$ , где  $S$  — максимальная длина строки.