

Построим по введенной строке суффиксный автомат. Будем для каждой вершины рассматривать самую длинную подстроку, которая с ней ассоциирована (именно длину этой строки мы храним в вершине). Для вершин введем величину *counter*, которая будет обозначать количество вхождений соответствующей строки в *S*. Изначально, *counter* равен 1 для вершин, не являющихся клонами.

Утверждение: если подстрока *T*, соответствующая вершине *v*, встречается в *S* *counter* раз, то для всех вершин на суффиксном пути из *v* (суффиксный путь получается последовательными переходами по суффиксным ссылкам из начальной вершины) мы должны прибавить *counter* к их счетчикам. Это следует из того, что для любой вершины *u* на суффиксном пути соответствующая строка *Q* является суффиксом *T*, и любое вхождение *T* содержит еще и вхождение *Q*. Однако, просто делать прибавления на суффиксном пути слишком долго (в худшем случае, это будет  $O(N)$ ).

Более быстрое решение использует отложенные прибавления *counter*. Упорядочим вершины по уменьшению длины соответствующей строки, и в этом порядке будем насчитывать *counter* для них. Когда мы приходим в очередную вершину, будем "пропихивать" *counter* по суффиксной ссылке (т.е. увеличивать ее *counter* на величину нашего). Остается только взять максимум длины по всем вершинам с  $K \leq counter$ .

Также эту задачу после построения суффиксного автомата можно решить за линейное время, избегая сортировки, вместо этого используя dfs по графу, образованному переходами по символам. Будет вычисляться та же величина *counter* путем суммирования значения ее у вершин, в которые ведут ребра из текущей (по аналогии, изначально *counter* будет инициализирован 1 для вершин не-клонов).