1 Суффиксное дерево

Рассмотрим бор, в который добавили все суффиксы строки. Эта структура содержит в себе всю информацию о подстроках и является мощным инструментом. Однако имеет размер $O(n^2)$.

Хочется ее как-то сжать. Для этого есть два основных направления.

- Можно заметить, что в нем $O(n)$ вершин у которых больше 1 сына и сжать пути из таких вершин в одно ребро
- Можно заметить, что у многих вершин поддерево устроено одинаково. Такие вершины можно объединить в одну.

Оба подхода приводят к структурам размера $O(n)$, строящися за линейное время. Мы рассмотрим второй подход.

2 Конечные автоматы

Конечным детерминированным автоматом назвается пятерка $(S, s, \Sigma, \delta, F)$, где

- S — конечное множество состояний.
- $s \in S$ — начальное состояние.
- Σ — конечный алфавит
- $\delta : S \times \Sigma \to S$ — функция переходов
- $F \subset S$ — множество терминальных состояний.

О таком автомате можно думать как об ориентированном графе, на каждом ребре которого написано по букве, причем их каждой вершины выходит не более одного ребра с такой буквой.

Будем говорить, что автомат принемает строку, если из начального состояния можно сделать последовательные переходы по буквам строки и попасть в терминальное состояние.

3 Суффиксный автомат

Суффиксным автоматом называется минимальный конечный детерминированный автомат, который принемает все суффиксы строки и только их.

Заметим, что в таком автомате нет циклов.

4 Правый контекст

Правым контекстом строки u относительно строки s (обозначаем $R_s(u)$) называется множество строк v, таких, что uv является суффиксом s.

Вершине автомата должен соответствовать в точности класс эквивалентности строк по правому контексту. Если в одну вершину автомата приведут строки с разным правым контекстом, то автомат некорректен, если несколько вершин автомата соотвествуют строкам с одинаковым правым контекстом, то не минимален.

5 Свойства правых контекстов

Теорема 1. *Пусть u — суффикс v. Тогда $R_s(v) \subset R_s(u)$.*

Теорема 2. *Пусть $R_s(u) \neq \varnothing, R_s(v) \neq \varnothing$, $|u| \leqslant |v|$, при этом $R_s(u) \cap R_s(v) \neq \varnothing$. Тогда $R(v) \subset R(u)$, причем u — суффикс v.*

Теорема 3. *Класс эквивалентности по равенству правых контекстов представляет собой несколько самых длинных суффиксов некоторой строки.*

Для непустой строки u определим $Suff(u)$ как самый длинный суффикс u, для которого правый контекст шире.

6 Изменение правых контекстов от добавления символа к строке s

Теорема 4. *При приписывании новой буквы к строке s*

- *К всем элементам правого контекста приписывается эта буква.*
- *Появляется новый класс эквивалентности с правым контекстом ε*
- *К правому контексту некоторых строк добавляется ε.*

Таким образом каждый класс, может разбиться не более чем на 2. На самом деле разобьется на 2 только один класс.

7 Изменение переходов между классами эквивалентности

Теорема 5. *Пусть $\varepsilon \notin R(u)$. Тогда переходы из этого класса, кроме перехода по новой не поменялись. Ни один из классов, в которые ведут переходы, кроме перехода по новой букве, не раздвоился. Если переход по новой букве раздвоился, то теперь переход ведет в половину без ε.*

Теорема 6. *Пусть $\varepsilon \in R(u)$, при этом их этого класса эквивалентности нет перехода по добавленной букве, тогда переходы из этого класса не поменялись. Ни один из классов, в которые они ведут не раздвоился, но добавился новый переход по добавленной букве в класс эквивалентности с правым контекстом $\{\varepsilon\}$*

Теорема 7. *Пусть $\varepsilon \in R(u)$, при этом их этого класса эквивалентности есть переход по добавленной букве, тогда переходы из этого класса, кроме перехода по новой не поменялись. Ни один из классов, в которые ведут переходы, кроме перехода по новой букве, не раздвоился. Если переход по новой букве раздвоился, то теперь переход ведет в половину с ε.*

Для того, чтобы класс раздвоился, необходимо чтобы у части строк появился ε в правом контексте, а у части нет. Но тогда, может раздвоиться только самый большого класса. Так как у остальных ε добавиться ко всем вершинам.

8 Алгоритм построения

Будем для каждой вершины хранить переходы, суффиксную ссылку, а также длину самой большой строки в классе эквивалентности.

Будем добавлять буквы по одной

- Создаем новую вершину, которой соответсвует класс $\{\varepsilon\}$.
- Идем по суффиксным ссылкам от новой вершины предыдущего шага, пока не упремся в корень, или не найдем переход по букве. По пути добавляем переходы в новую вершину.
- Если дошли до корня — он суффиксная ссылка новой вершины, шаг закончен
- Если нет, то надо понять раздвоится ли вершина. Если нет, то она суффиксная ссылка новой вершины, и шаг закончен.

- Если раздвоится, то надо склонировать вершины, пойти дальше вдоль суффиксного пути, и все переходы в старую вершину, перенаправить в клона. При этом клон становится суффиксной ссылкой и новой вершины, и вершины с которой он склонирован.

Для понимания раздвоится ли вершина, будкм хранить самую длинную строку в классе эквивалентности. Вершина раздвоится если самая длинная строка в ней, более чем на 1 длинее самой длинной там, откуда переход.

9 Доказательство линейности

Следим за $Len(Suff(Suff(new)))$. Так получается время работы $O(n * \Sigma)$.

10 Примеры задач

1. Добавлять символ, проверять подстрока ли
2. Идти строкой и поддерживать самый большой суффикс, который является подстрокой строки
3. Найти количество подстрок
4. Найти количество подстрок у всех префиксов
5. Найти подстроку с максимальным количеством вхождений, среди таких самую длинную.