Алгоритм Маккрейта

Выполнила:
Белкова Елизавета

Студентка Дальневосточного федерального университета направления Прикладная информатика 2 курса

Общая идея

Алгоритм Маккрейта (англ. *McCreight's algorithm)*

- построение суффиксного дерева для заданной строки
- добавляет суффиксы в порядке убывания их длины
- за линейное время

Введение

- Функция поиска определенной подстроки в более длинной основной строке часто используется во многих приложениях
- В таком случае лучше использовать алгоритмы, в которых скорость поиска осуществляется за линейное время
- В основном эти алгоритмы базируются на суффиксных деревьях.

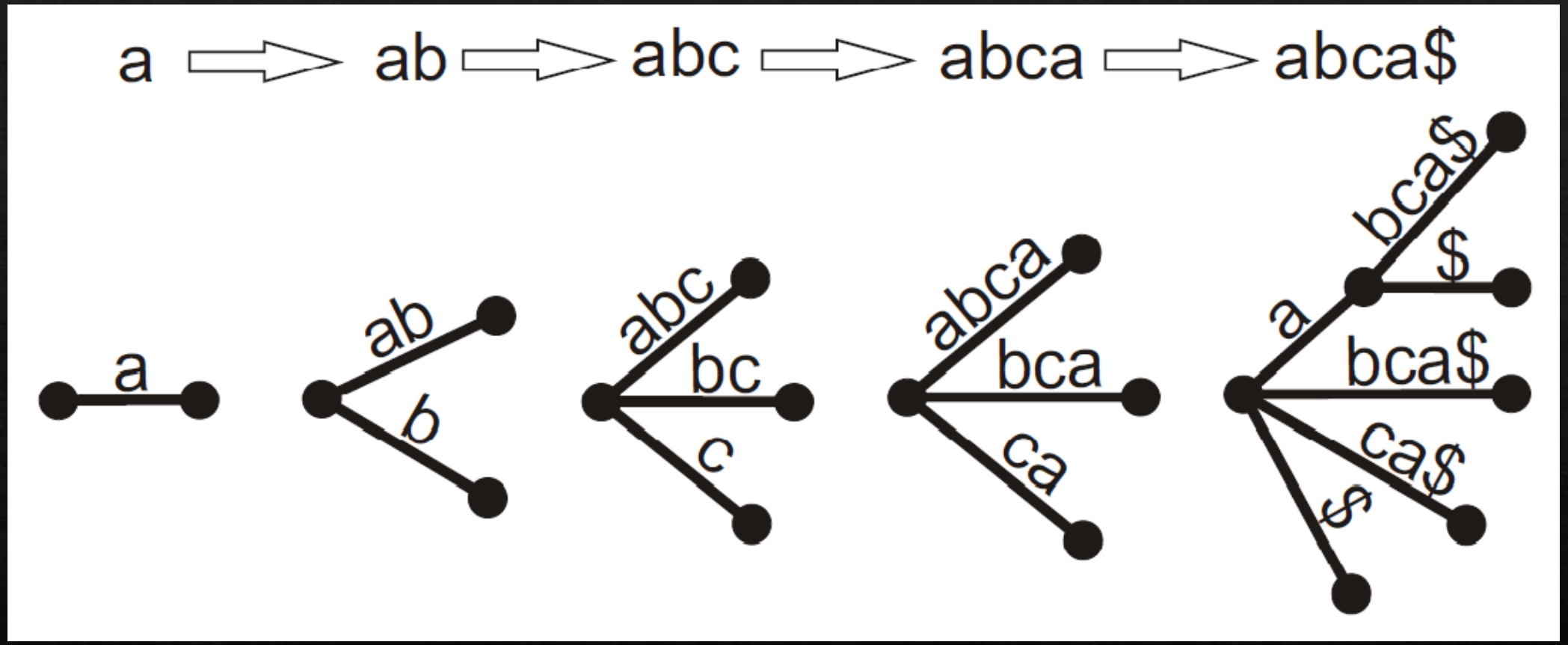
История алгоритма

- Алгоритм Кнута-Морриса-Пратта (1970 год)
- Алгоритм Вайнера (1973 год)
- Алгоритм Маккрейта (1976 год)
- Алгоритм Укконена (1995 год)

Сферы применения

- Редактирование текста
- Браузерный поиск
- Вычислительная биология (структуры ДНК)

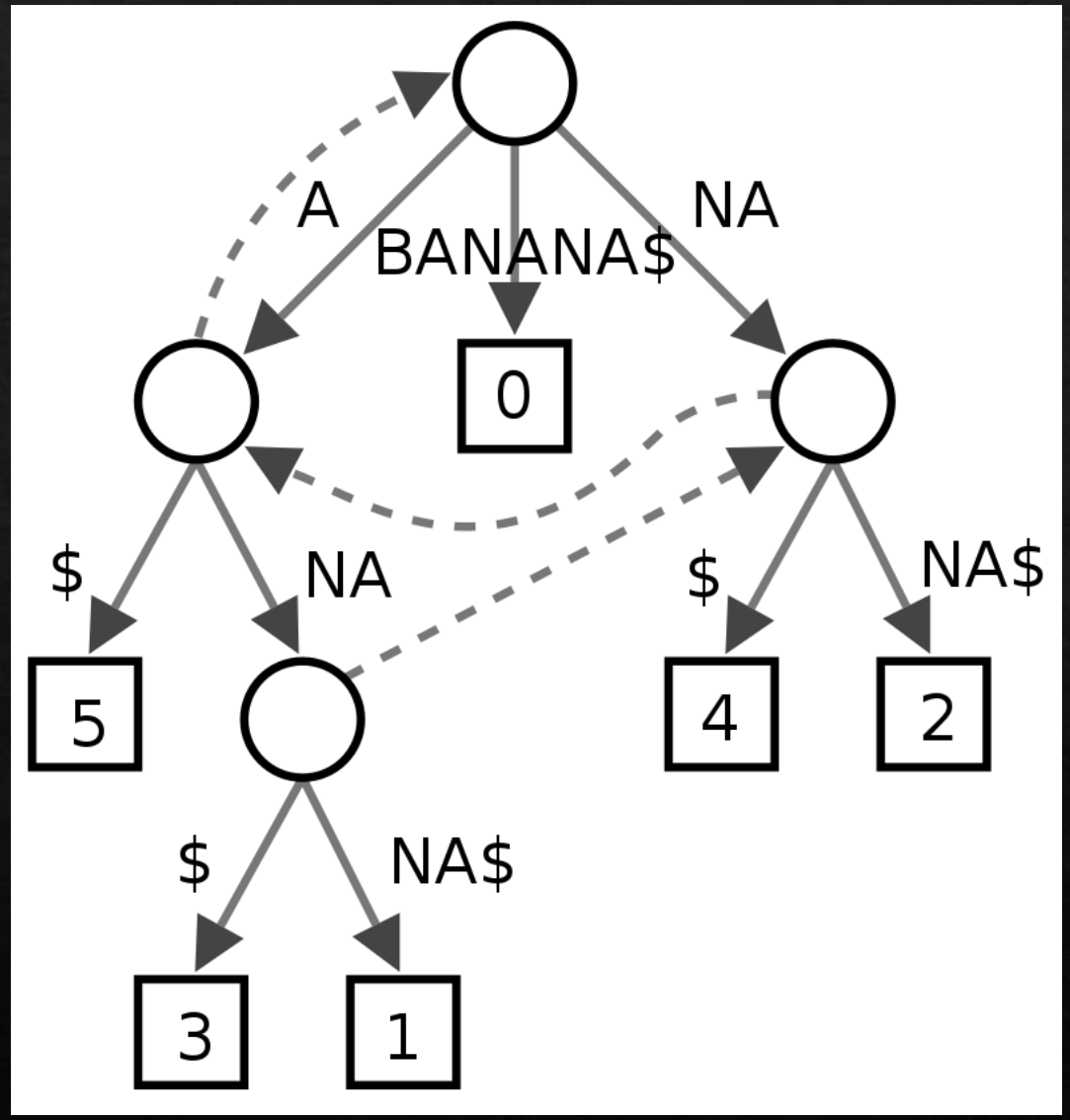
- Алгоритм Маккрейта строит *суффиксное дерево* из заданной строки
- Суффиксное дерево (Suffix Tree) – это структура данных, содержащая все суффиксы некоторой входной строки



Пример построения суффиксного дерева

Маккрейт в 1976 году предложил свой алгоритм, в котором порядок добавления суффиксов – от большего к меньшему.

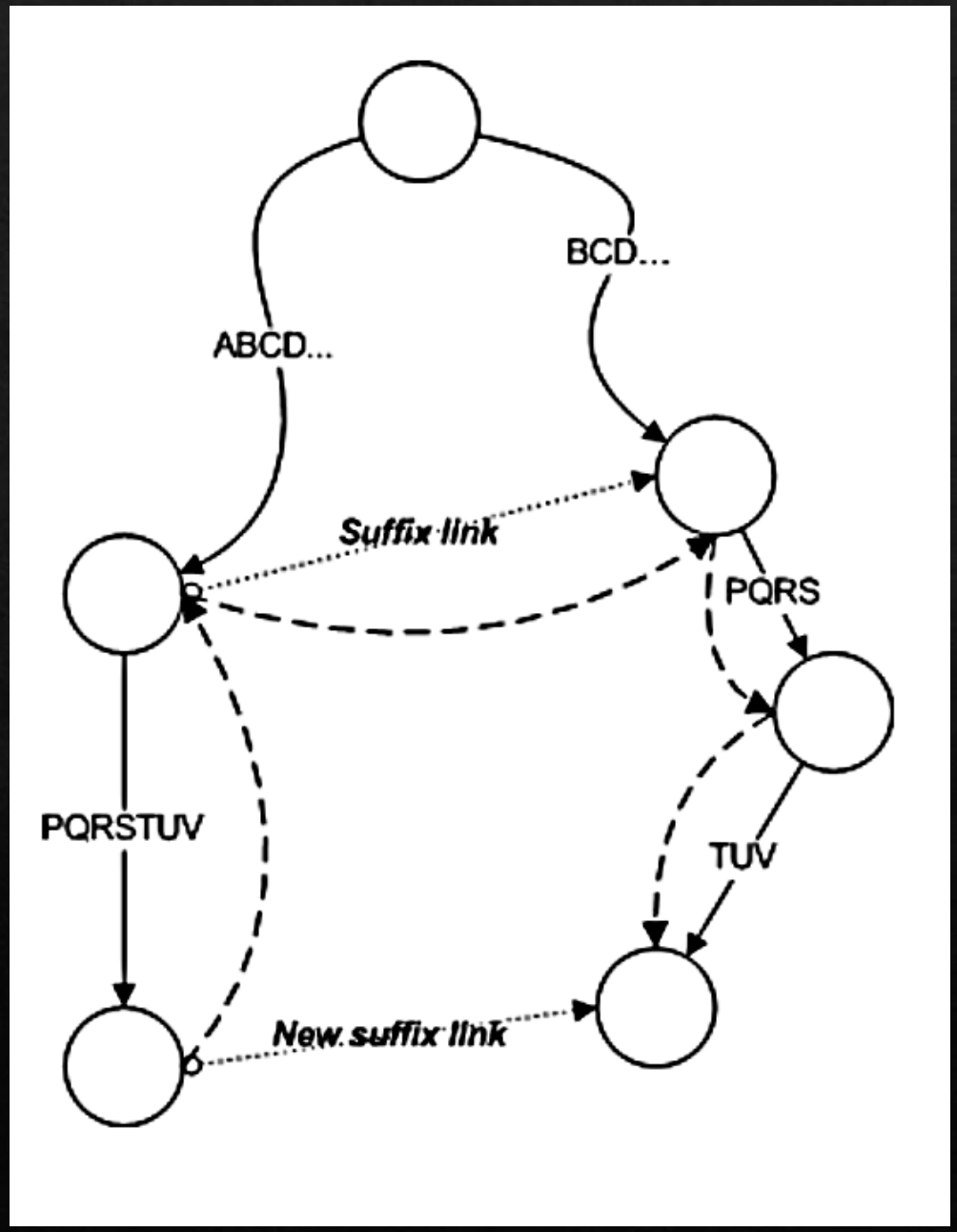
Для быстрого вычисления места, откуда нужно продолжить построение нового суффикса, достаточно *суффиксной ссылки* в каждой вершине

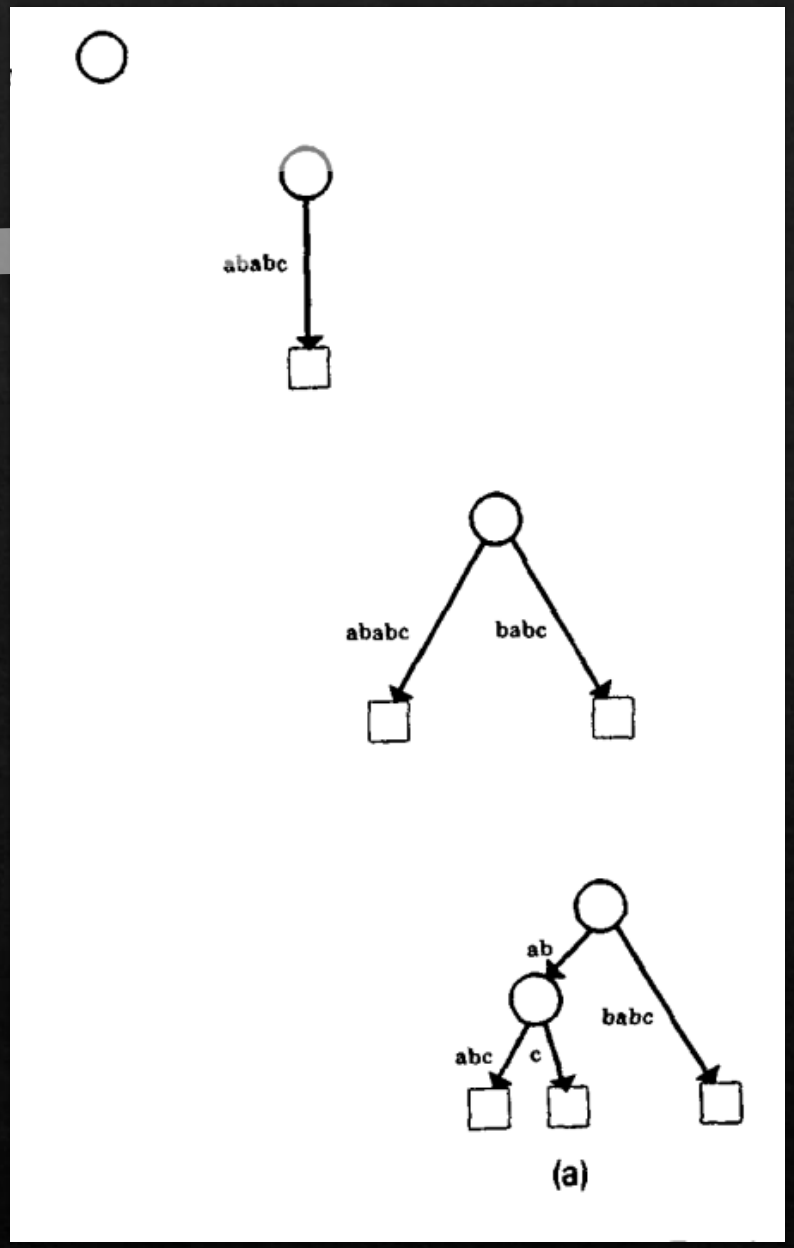


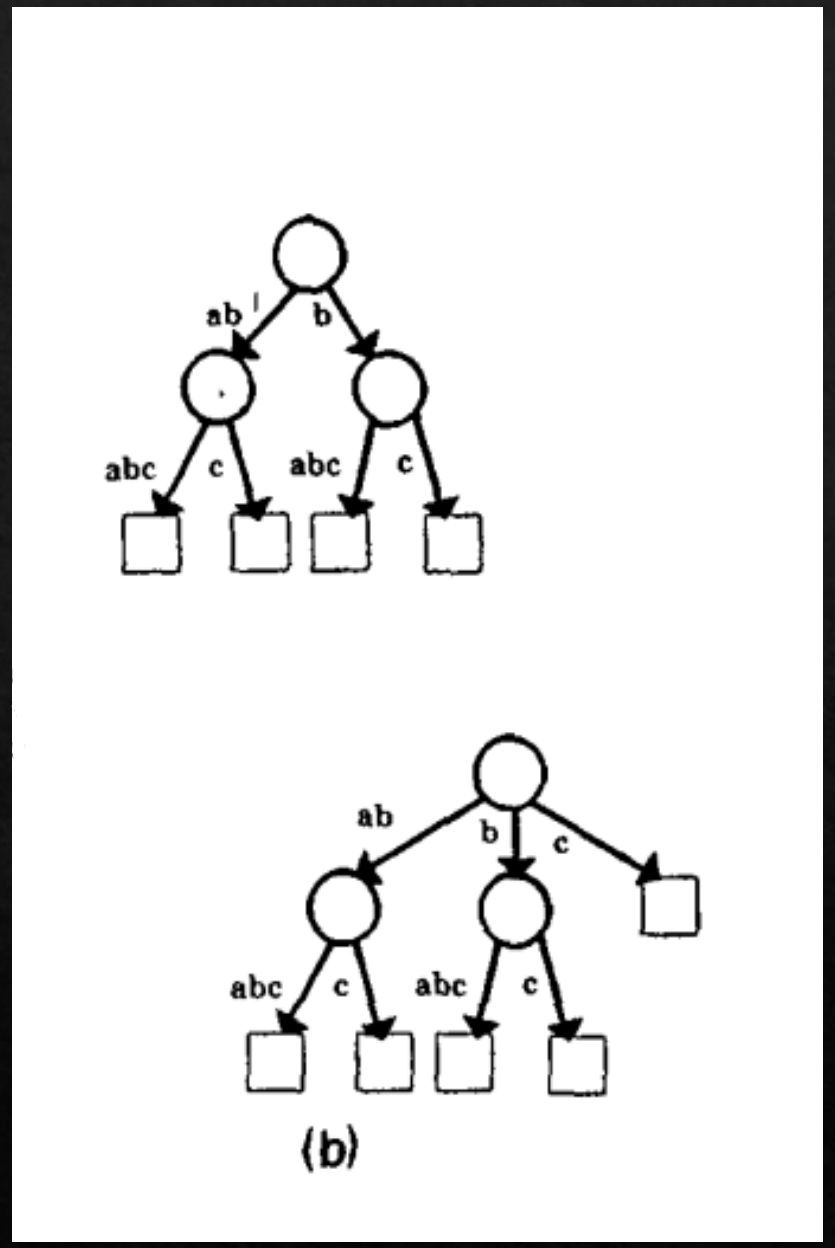
Пример дерева с суффиксными ссылками

Построение суффиксного дерева Маккрейта

— алгоритм, который вычисляет суффиксные ссылки во время построения суффиксного дерева и использует их как короткие пути







Пример построения суффиксного дерева с помощью алгоритма Маккрейта

Реализация в коде

Схема алгоритма Маккрейта:

T := two-node tree with one edge labeled by $p_1 = S$;

for $i := 2$ **to** n **do begin**

{ insert next suffix $p_i = S[i. .n]$ }

localize $head_i$ as $head(p_i, T)$,

starting the search from suf [father($head_i$ -1)]

and using *fastfind* whenever possible;

T := insert(p_i ,T);

end

Схема алгоритма Маккрейта с использованием суф. ссылок:

T := two-node tree with one edge labeled by $p_1 = S$;

for $i := 2$ **to** n **do begin**

{ insert next suffix $p_i = S[i. .n]$ }

let (β be the label of the edge (father[$head_{i-1}$], $head_{i-1}$);

let γ be the label of the edge ($head_{i-1}$, $leaf_{i-1}$);

u := suf [father[$head_{i-1}$]];

v := *fastfind*(u, β);

suf [$head_{i-1}$] := v;

if v has only one son **then**

{ v is a newly inserted node } $head_i := v$

else $head_i$:= *slowfind*(v,γ);

create a new leaf $leaf_i$, make $leaf_i$ a son of $head_i$;

label the edge ($head_i$, $leaf_i$) accordingly;

end

Асимптотическая оценка

- В приведенном алгоритме используется константное число операций на добавление одного суффикса.
- Итоговая асимптотика алгоритма O(N)
- Алгоритм линейный

```
Длина исходной строки: 11
Исходная строка со знаком $: abcdeeeeaaa$
Время построения суффиксного дерева: 0.009 seconds
```

```
Длина исходной строки: 5
Исходная строка со знаком $: abcde$
Время построения суффиксного дерева: 0.005 seconds
```

```
Длина исходной строки: 9
Исходная строка со знаком $: breakdown$
Время построения суффиксного дерева: 0.006 seconds
```

```
Длина исходной строки: 11
Исходная строка со знаком $: abracadabra$
Время построения суффиксного дерева: 0.009 seconds
```

Сравнение с алгоритмом Вайнера

В сравнении с алгоритмом Вайнера:

Преимущества: каждая вершина хранит только суффиксную ссылку, а не массивы размера алфавита.

Недостатки: нет

Сравнение с алгоритмом Укконена

В сравнении с алгоритмом Укконена:

Преимущества: мы строим суффиксное дерево в явной форме, что может облегчить понимание алгоритма.

Недостатки: является offline алгоритмом, то есть требует для начала работы всю строку целиком.