

# Алгоритм Бойера - Мура

Применяется для поиска  
подстроки в строке

# История создания

- Алгоритм поиска строки Бойера — Мура, считается наиболее быстрым среди алгоритмов общего назначения, предназначенных для поиска подстроки в строке. Был разработан Робертом Бойером (англ.)русск. и Джейем Муром в 1977 году.

# Основные идеи алгоритма

- Сканирование слева направо, сравнение справа налево
- Поиск стоп - символа
- Поиск совпавшего суффикса

# Сканирование и сравнение

- Совмещается начало строки и начало шаблона, проверка идет с последнего символа шаблона
- Если символы совпадают, то производится сравнение предпоследнего символа шаблона и т.д.
- Если все символы совпали, то образец найден

# СТОП - СИМВОЛ

- Если с шаблоном не совпала первая сравниваемая буква, то сдвигаем шаблон вправо до последней такой же буквы
- Если в шаблоне нет стоп – символа, то сдвигаем шаблон за стоп – символ.

# Стоп - символ

- Предположим, что мы производим поиск слова «колокол». Первая же буква не совпала — «к» (назовём эту букву *стоп-символом*). Тогда можно сдвинуть шаблон вправо до *последней* его буквы «к».

```
Строка:      * * * * * * к * * * * * *
Шаблон:      к о л о к о л
Следующий шаг:      к о л о к о л
```

- Если стоп-символа в шаблоне вообще нет, шаблон смещается за этот стоп-символ.

```
Строка:      * * * * * а л * * * * * *
Шаблон:      к о л о к о л
Следующий шаг:      к о л о к о л
```

# Стоп - символ

- В данном случае стоп-символ — «а», и шаблон сдвигается так, чтобы он оказался прямо за этой буквой. В алгоритме Бойера-Мура эвристика стоп-символа вообще не смотрит на совпавший суффикс, так что первая буква шаблона («к») окажется под «л», и будет проведена одна заведомо холостая проверка.
- Если стоп-символ «к» оказался за другой буквой «к», эвристика стоп-символа не работает.

```
Строка:      * * * * к к о л * * * * *
Шаблон:      к о л о к о л
Следующий шаг: к о л о к о л      ??????
```

# Суффикс

- Если при сравнении строки и шаблона совпало 1 или больше символов, то шаблон сдвигается в зависимости от того, какой суффикс совпал

```
Строка:      * * Т О К О Л * * * * *
Шаблон:      К О Л О К О Л
Следующий шаг:      К О Л О К О Л
```

- В данном случае совпал суффикс «окол», и шаблон сдвигается вправо до ближайшего «окол». Если подстроки «окол» в шаблоне больше нет, но он начинается на «кол», сдвигается до «кол», и т. д.

# Таблица стоп - СИМВОЛОВ

- В таблице указывается последняя позиция элемента в шаблоне (за исключением последней буквы)
- Если в шаблоне нет такого элемента то в таблицу записывается ноль

# Таблица суффиксов

- Для каждого возможного суффикса в таблицу записывается наименьшая величина, на которую надо сдвинуть шаблон чтобы он снова совпал с суффиксом

# Достоинства алгоритма

- Оптимален при отсутствии возможности провести предварительную обработку текста
- Достаточно быстрый в большинстве случаев

# Недостатки алгоритма

- На больших алфавитах таблица стоп – символов может занимать много памяти
- На некоторых “неудачных” текстах его скорость сильно снижается

**Спасибо за внимание!**