# Boyer Moore Algorithm for Pattern Searching

<https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/>

## 1.用于模式搜索的博耶摩尔算法

模式搜索是计算机科学中的一个重要问题。

当我们在记事本/Word 文件、浏览器或数据库中搜索字符串时，

会使用模式搜索算法来显示搜索结果。

典型的问题是

Given a text txt[0..n-1]

and a pattern pat[0..m-1]

where n is the length of the text

and m is the length of the pattern,

write a function search(char pat[], char txt[])

that prints all occurrences of pat[] in txt[].

You may assume that n > m.

**Examples:**

**Input:** txt[] = “THIS IS A TEST TEXT”

pat[] = “TEST”

**Output:** Pattern found at index 10

**Input:** txt[] = “AABAACAADAAB**AABA**”

pat[] = “AABA”

**Output:** Pattern found at index 0

Pattern found at index 9

Pattern found at index 12

In this post, we will discuss the Boyer Moore pattern searching algorithm.

Like [KMP](https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/) and [Finite Automata](https://www.geeksforgeeks.org/finite-automata-algorithm-for-pattern-searching/) algorithms, Boyer Moore algorithm also preprocesses the pattern.

与KMP和有限自动机算法一样，Boyer Moore算法也对模式进行预处理。

Boyer Moore is a combination of the following two approaches.

1. **Bad Character Heuristic**
2. **Good Suffix Heuristic**

坏字符启发法

好的后缀启发式

Both of the above heuristics can also be used independently to search a pattern in a text.

Let us first understand how two independent approaches work together in the Boyer Moore algorithm.

If we take a look at the [Naive algorithm](https://www.geeksforgeeks.org/naive-algorithm-for-pattern-searching/), it slides the pattern over the text one by one. [KMP algorithm](https://www.geeksforgeeks.org/searching-for-patterns-set-2-kmp-algorithm/) does preprocessing over the pattern so that the pattern can be shifted by more than one. The Boyer Moore algorithm does preprocessing for the same reason. It processes the pattern and creates different arrays for each of the two heuristics. At every step, it slides the pattern by the max of the slides suggested by each of the two heuristics. **So, it uses greatest offset suggested by the two heuristics at every step**.

如果我们看一下Naive算法，它会在文本上一个接一个地滑动模式。

KMP算法对模式进行预处理，使模式可以移位多个。

Boyer Moore算法也出于同样的原因进行预处理。

它处理模式并为两种启发式方法中的每一种创建不同的数组。

在每一步中，它都以两种启发式方法所建议的最大滑动次数滑动模式。

因此，它在每一步都使用两种启发式建议的最大偏移量。

Unlike the previous pattern searching algorithms, the **Boyer Moore algorithm starts matching from the last character of the pattern.**  
In this post, we will discuss the bad character heuristic and the Good Suffix heuristic in the next post.

与以前的模式搜索算法不同，Boyer Moore算法从模式的最后一个字符开始匹配。

在这篇文章中，我们将在下一篇文章中讨论坏字符启发式和好后缀启发式。

### ****Bad Character Heuristic****

The idea of bad character heuristic is simple.

The character of the text which doesn’t match with the current character of the pattern is called the **Bad Character**.

Upon mismatch, we shift the pattern until –

1. The mismatch becomes a match.
2. Pattern P moves past the mismatched character.