

即是暴力（**BruteForce**）算法，是普通的模式匹配算法，**BF**算法的思想就是将目标串**T**的第一个字符和主串的**S**的第一个字符进行匹配，若相等，则继续匹配**T**串和**S**串的第二个字符，若不相等，则比较主串**S**的第二个字符和模式串**T**的第二个字符，依次比较下去，知道得到最后的匹配结构。**BF**算法也是一种蛮力算法。

算法思想：

普通模式匹配算法，其实现过程没有任何技巧，就是简单粗暴地拿一个串同另一个串中的字符一一比对，得到最终结果。

代码复杂度： $O(n*m)$

该算法最理想的时间复杂度  $O(n)$ ， $n$  表示串  $A$  的长度，即第一次匹配就成功。**BF** 算法最坏情况的时间复杂度为  $O(n * m)$ ， $n$  为串  $A$  的长度， $m$  为串  $B$  的长度。例如，串  $B$  为 “0000000001”，而串  $A$  为 “01”，这种情况下，两个串每次匹配，都必须匹配至串  $A$  的最末尾才能判断匹配失败，因此运行了  $n * m$  次。

演示过程：

使用普通模式匹配算法判断串  $A$  (“abcac”) 是否为串  $B$  (“ababcabcacbab”) 子串的判断过程如下：

1. 首先，将模式串和主串对齐，依次进行匹配是否相等

B: ababcabcacbab

A: abcac

2. 若不相等， $B$ 串开始回溯，从第二个字符开始与 $A$ 串的第一个字符开始匹配是否相等

B: ababcabcacbab

A: abcac

3. 若不相等， $B$ 串开始回溯，从第三个字符开始与 $A$ 串的第一个字符开始匹配是否相等

B:ababcbacbab

A: abcac

4. 两串的模式匹配失败，串 A 继续移动，一直移动至匹配的位置才匹配成功：

B:ababcbacbab

A: abcac

由此，串 A 与串 B 以供经历了 6 次匹配的过程才成功，通过整个模式匹配的过程，证明了串 A 是串 B 的子串（串 B 是串 A 的主串）。

## BF算法实现-C语言

```
1  #include <stdio.h>
2  #include <string.h>
3  int BF(char *S,char *T,int pos)
4  {
5      int i = pos;//开始匹配的位置
6      int j = 0;
7      if (pos<1 || pos>strlen(S))
8      {
9          printf("匹配位置有误! ");
10         return 0;
11     }
12     while (i<=strlen(S) && j<=strlen(T))
13     {
14         if (S[i - 1] == T[j])
15         {
16             ++i;
17             ++j;
18         }
19         else
20         {
21             i = i - j + 1;
22             j = 0;
23         }
24     }
25     if (j == strlen(T))
26         return i - strlen(T);
27     else
28         return 0;
29 }
30 int main()
31 {
32     printf("BF算法：模式串T在主串S中的位置是： %d" ,BF("HELLO WORLD program", "gram",
33     1));
34     return 0;;
35 }
```

总结：

BF算法的时间复杂度很高，也是一种蛮力的模式匹配算法，算法效率很低。