BF（Brute Force）算法核心思想是：首先S[1]和T[1]比较，若相等，则再比较S[2]和T[2]，一直到T[M]为止；若S[1]和T[1]不等，则T向 右移动一个字符的位置，再依次进行比较。如果存在k，1≤k≤N，且S[k+1…k+M]=T[1…M]，则匹配成功；否则失败。该算法最坏情况下要进行 M\*(N-M+1)次比较，[时间复杂度](http://baike.baidu.com/view/104946.htm" \t "_blank)为O(M\*N)

     比如: String a ="123456789" String b= "23"  
   其实就是循环 a一个一个字符匹配，开始a[0]和b[0]匹配，没有找到，直接不用找b[1]，直接用a[1]和b[0]匹配发现已经 找到，然后继续用

a[2]和b[1]匹配发现已经 找到。前且是b字符串末尾。说明已经完全找到。返回结果

**[java]** [view plain](http://blog.csdn.net/zjcheer_up/article/details/37959029) [copy](http://blog.csdn.net/zjcheer_up/article/details/37959029)

1. **package** com.string;
2. //BF算法
3. /\*
4. \* 1.问题：找出str2在str1中出现的起始位置
5. \* \*/
6. **public** **class** BF01 {
7. **public** **static** **void** main(String args[]){
8. String str1 = "ailkmno";
9. String str2 = "ilkm";
10. **int** n = find(str1,str2);
11. System.out.println(str2+"在"+str1+"中出现的位置是："+n);
13. }
14. **public** **static** **int** find(String str1,String str2){
15. **char**[] c1 = str1.toCharArray();
16. **char**[] c2 = str2.toCharArray();
17. **int** i=0;
18. **int** j;
19. **while**(i<str1.length()-1){
20. j = 0;
21. **while**(c1[i]==c2[j]&& j< str2.length()-1){
22. i++;
23. j++;
24. }
25. **if**(j==str2.length()-1){
26. **return** i - str2.length()+1;
28. }
29. i=i-j+1;
31. }
32. **return** -1;
33. }

36. }

**[java]** [view plain](http://blog.csdn.net/zjcheer_up/article/details/37959029) [copy](http://blog.csdn.net/zjcheer_up/article/details/37959029)

1. **package** com.test;
2. //BF算法
3. /\*
4. \* 1.问题：找出str2在str1中出现的起始位置
5. \* \*/
7. **public** **class** Test2 {
8. **public** **static** **void** main(String args[]){
9. String str1 = "ailkmno";
10. String str2 = "ilkm";
11. **int** n = find(str1,str2);
12. System.out.println(str2+"在"+str1+"中出现的位置是："+n);
14. }
15. **public** **static** **int** find(String str1,String str2){
16. **char**[] c1=str1.toCharArray();
17. **char**[] c2=str2.toCharArray();
18. **int** j=0;
19. **int** i=0;
20. **for**(;i<str1.length();){
21. **if**(j>=str2.length()-1){
22. **break**;
23. }
24. **if**(c1[i]==c2[j] ){
25. i=i+1;
26. j=j+1;
27. }**else**{
28. i=i+1;
29. }
30. }
31. System.out.println(i+"                 "+j);
32. **return** i-str2.length()+1;
33. }
34. }