

## # 字符串匹配

### ## 字符串匹配问题

#### ### 单串匹配

一个模式串(pattern) 一个待匹配串，找出前者在后者中的所有出现位置。

#### ### 多串匹配

多个模式串，一个待匹配串（多个待匹配串可以直接连起来）。

直接当作单串匹配肯定是可以的，但是效率不够高。

#### ### 匹配一个串的任意后缀

#### ### 匹配多个串的任意后缀

### ## 暴力做法

对于每个位置，尝试对模式串和待匹配串进行比对。

参考代码：

（伪代码）

```
```c++ std::vector<int> match(char *a, char *b, int n, int m) {
```

```
    std::vector<int> ans;
    for (i = 0; i < n - m + 1; i++) {
        for (j = 0; j < m; j++) {
            if (a[i + j] != b[j]) break;
        }
        if (j == m) ans.push_back(i);
    }
    return ans;
```

```
} ```
```

时间复杂度分析：

最坏时间复杂度是  $O(nm)$  的，

最好是  $O(n)$  的。

如果字符集的大小大于 1（有至少两个不同的字符），平均时间复杂度是  $O(n)$  的。但是在 OI 题目中，给出的字符串一般都不是纯随机的。

### ## Hash 的方法

### ## KMP 算法

### ## 参考链接

[Oi Wiki](<https://oi-wiki.org/string/match/>)

Last  
update: 2020-2021:teams:legal\_string:字符串 https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\_string:%E5%AD%97%E7%AC%A6%E4%B8%B2%E5%8C%B9%E9%85%8D\_lgwza&rev=1594809821  
2020/07/15 匹配\_lgwza  
18:43

---

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\\_string:%E5%AD%97%E7%AC%A6%E4%B8%B2%E5%8C%B9%E9%85%8D\\_lgwza&rev=1594809821](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:%E5%AD%97%E7%AC%A6%E4%B8%B2%E5%8C%B9%E9%85%8D_lgwza&rev=1594809821)

Last update: **2020/07/15 18:43**

