

# BAT算法面试题(二) - 无重复字符的最长子串

---

## 一.算法题

---

- 题目

Given a string, find the length of the longest substring without repeating characters.

- Example

- Given "abcabcbb", the answer is "abc", which the length is 3.
- Given "bbbbbb", the answer is "b", with the length of 1.
- Given "pwwkew", the answer is "wke", with the length of
- Note that the answer must be a substring, "pwke" is a subsequence and not a substring.

## 二.算法题解读

---

- 题目大意:给定一个字符串,找出不含有重复字符的最长子串的长度

- 解读Example

- 给定"abcabcbb",没有重复字符的最长子串是"abc",那么长度就是3
- 给定"bbbbbb",最长子串就是"b",长度就是1
- 给定pwwkew,最长子串就是"wke",长度为3,
- **注意**,必须是一个子串."pwke",是子序列,而不是子串

## 三.暴力解决方案

---

### 3.1 思路

逐个检查所有的子字符串,看它是否不含有重复字符

## 3.2 算法

为了枚举给定字符串的所有子字符串,我们需要枚举它们开始和结束的索引,假如开始和结束的索引分别是*i*和*j*.那么我们有  $0 \leq i \leq j \leq n$ . 因此,使用 *i*从0到*n*-1 以及 *j*从*i*+1到*n* 这2个嵌套循环.我们就可以遍历出 *a* 的所有子字符串.

## 3.3 复杂的分析

- 时间复杂度:  $O(n^3)$  ;
- 空间复杂度:  $O(\min(n, m))$  ;

## 3.4 参考代码

```
//(2)无重复字符的最长子串
//求字符串长度函数
int strLength(char *p)
{
    int number = 0;
    while (*p) {
        number++;
        p++;
    }
    return number;
}

//判断子字符在字符串中是否唯一
int unRepeatStr(char *a, int start, int end)
{
    for (int i = start; i < end; i++) {
        char c = a[i];
        for (int j = i + 1; j < end; j++) {
            if (c == a[j]) {
                return 0;
            }
        }
    }
    return 1;
}
```

*//找出不含有重复字符串最长子串的长度*

```
int LengthLongestSubstring(char *a)
{
    int n = strlen(a);
    int ans = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (unRepeatStr(a, i, j)) {
                ans = (ans > j-i)?ans:j-i;
            }
        }
    }

    return ans;
}

int main(int argc, const char * argv[]) {

    //2)无重复子串的最长子串
    char *s = "pwwkew";
    int n = LengthLongestSubstring(s);
    printf("%d",n);

    return 0;
}
```