

*KMP*算法原理和代码详解

前置知识

无

*KMP*算法的内容分为两期

本期为 *KMP*算法原理和代码详解

下期为 *KMP*算法相关题目

这一定是*KMP*算法讲解最清晰的视频

KMP算法原理和代码详解

$s1$ 字符串是否包含 $s2$ 字符串，如果包含返回 $s1$ 中包含 $s2$ 的最左开头位置，不包含返回 -1

暴力方法就是 $s1$ 的每个位置都做开头，然后去匹配 $s2$ 整体，时间复杂度 $O(n * m)$

KMP算法可以做到时间复杂度 $O(n + m)$

KMP算法详解

- 1) 理解 $next$ 数组的定义，定义是一切的关键，前缀和后缀的最大匹配长度
- 2) 假设已经有了 $next$ 数组，详解匹配过程是如何得到加速的，加速过程有2个理解核心
- 3) 理解了匹配主流程之后，详解 $next$ 数组如何快速生成，不停跳跃的过程有1个理解核心
- 4) KMP算法代码详解，主流程 + $next$ 数组生成
- 5) 时间复杂度 $O(n)$ 的证明，直接从代码层次就可以分析出来，分析方式好理解，但是比较特别
我们图解一下，例子会很详细

以后可以认为，求解1个位置的 $next$ 数组的值，时间复杂度 $O(1)$

很多题目中都有 $next$ 数组的使用，具体请看下期 - KMP算法相关题目

*KMP*算法原理和代码详解

题目 1

*KMP*算法的实现

时间复杂度 $O(n + m)$

KMP算法原理和代码详解

题目2

另一棵树的子树

给你两棵二叉树 $root$ 和 $subRoot$

检验 $root$ 中是否包含和 $subRoot$ 具有相同结构和节点值的子树

如果存在，返回 $true$

否则，返回 $false$

测试链接：<https://leetcode.cn/problems/subtree-of-another-tree/>

本题需要理解二叉树先序序列化，讲解036，题目5

时间复杂度 $O(n + m)$