

数据结构与算法

Data Structure and Algorithm

XXIX 字符串

授课人: Kevin Feng

翻 译: 赵 其 辰

课前回顾



数学回顾



数组



递归 vs 迭代



动态数组 vs 链表



堆栈和队列



哈希表



树



堆



排序



图论



分治法



动态规划



字符串所用技巧

- ◎ 排序
- ◎ 递归
- ◎ 二分法搜索
- ◎ 分治算法
- ◎ 双指针
- ◎ 滑动窗口
- ◎ 动态规划
- ◎ 贪婪算法
- ◎ 使用各种数据结构

字符串题型

- ◎ 基于字符计数的问题

- ◎ 同字母异序

- ◎ 回文

- ◎ 二进制字符串

- ◎ 子序列

- ◎ 模式搜索

- ◎ 其他

计数问题 I

- ◎ 偶数子串的数量

- ◎ 给定一串0到9的数字。任务是计算在将整数转换为偶数时的子串的数量。

- ◎ Input : str = "1234".

- ◎ Output : 6

- ◎ “2”， “4”， “12”， “34”， “234”， “1234是6个子字符串，它们是偶数。

计数问题 II

◎ 学生出勤记录

- ◎ 给你一个代表学生出勤记录的字符串。 该记录只包含以下三个字符：
 - ◎ 'A' : 缺席.
 - ◎ 'L' : 迟到.
 - ◎ 'P' : 出席.
- ◎ 如果学生的出勤记录不包含多于一个“A”（缺席）或超过两个连续的“L”（迟到），则可以获得奖励。
- ◎ 你需要根据他的出勤记录来返回学生是否可以得到奖励。

计数问题 III

- ◎ 对具有相同首尾字符的子字符串进行计数
- ◎ 给出一个字符串S，我们需要找到所有连续的子字符串开始和结束的字符都相同的计数。
- ◎ Input : S = "abcab"
- ◎ Output : 7
- ◎ "abcab" 有15个子字符串
- ◎ 在上面的子串中，有7个子串满足要求：a, abca, b, bcab, c, a和b。.

计数问题 IV

- ◎ 字符串中最大连续重复字符

- ◎ 给定一个字符串，其任务是在字符串中查找最大连续重复字符。

- ◎ 排序数组中删除重复

字符串题型

- ◎ 基于字符计数的问题
- ◎ 同字母异序
- ◎ 回文
- ◎ 二进制字符串
- ◎ 子序列
- ◎ 模式搜索
- ◎ 其他

同字母异序词 I

◎ 同字母异序词

- ◎ 编写一个函数来检查两个给定的字符串是否彼此是相同字母异序词。 一个字符串的相同字母异序词是另一个包含相同字符的字符串，只有字符的顺序可以不同。 例如，“abcd”和“dabc”是彼此的相同字母异序词。

同字母异序词 II

◎ 查找字符串中的所有相同字母异序词

- ◎ 给定字符串s和非空字符串p, 在s中找到所有p的相同字母异序词的起始索引。
- ◎ 字符串只包含小写英文字母, 字符串s和p的长度不会超过20, 100。
- ◎ 输出的顺序并不重要。
- ◎ Input: s: "cbaebabacd" p: "abc"
- ◎ Output: [0, 6]
- ◎ 解释:
 - ◎ 以0作为起始索引 的子序列是 "cba", 它是"abc"的相同字母异序词.
 - ◎ 以6作为起始索引 的子序列是 "bac", 它是"abc"的相同字母异序词.

同字母异序词 III

◎ 查找同字母异序词的映射

- ◎ 给定两个列表A和B，B是A的一个同字母组。这意味着B是通过随机化A中元素的顺序而创建的。
- ◎ 我们希望找到一个从A到B的索引映射P。映射 $P[i] = j$ 意味着列表A中的第i个元素出现在B的索引为j的位置。
- ◎ 这些列表A和B可能包含重复项。如果有多个答案，则输出它们中的任意一个。
- ◎ 例如，给定
 - ◎ $A = [12, 28, 46, 32, 50]$
 - ◎ $B = [50, 12, 32, 46, 28]$
- ◎ 应当返回 $[1, 4, 3, 2, 0]$
- ◎ $P[0] = 1$ 因为A的第0个元素出现在B[1]处，并且 $P[1] = 4$ ，因为A的第1个元素出现在B[4]处，依此类推。

字符串题型

- ⊙ 基于字符计数的问题
- ⊙ 同字母异序
- ⊙ 回文
- ⊙ 二进制字符串
- ⊙ 子序列
- ⊙ 模式搜索
- ⊙ 其他

回文 I

◎ 移位

- ◎ 给定两个字符串s1和s2，写一段程序说明s2是否是s1 的移位。

◎ 移位 II

- ◎ 写一个函数 `rotate(arr[], d, n)` 将大小为n的数组arr[] 移位d个单位。

◎ 回文数

- ◎ 判断一个整数是否是回文数。当一个整数从前从后读一样时可判定为回文数。

回文 II

- ◎ 移位回文

- ◎ 检查给定的字符串是否是一个回文字符串的移位。

- ◎ 重排回文

- ◎ 给定一个字符串，检查字符串中的各字符是否可以构成一个回文字符串。

- ◎ 最长回文

- ◎ 给定一个由大小写字母组成的字符串，找到可由这些字符构成的最长的回文字符串。
- ◎ 本例是大小写敏感的，例如“Aa”在此处不被认为是回文字符串。

回文 III

◎ 回文流判定

```
Input: str[] = "abcba"
```

```
Output: a Yes    // "a" is palindrome  
        b No     // "ab" is not palindrome  
        c No     // "abc" is not palindrome  
        b No     // "abcb" is not palindrome  
        a Yes    // "abcba" is palindrome
```

```
Input: str[] = "aabaacaabaa"
```

```
Output: a Yes    // "a" is palindrome  
        a Yes    // "aa" is palindrome  
        b No     // "aab" is not palindrome  
        a No     // "aaba" is not palindrome  
        a Yes    // "aabaa" is palindrome  
        c No     // "aabaac" is not palindrome  
        a No     // "aabaaca" is not palindrome  
        a No     // "aabaacaa" is not palindrome  
        b No     // "aabaacaab" is not palindrome  
        a No     // "aabaacaaba" is not palindrome  
        a Yes    // "aabaacaabaa" is palindrome
```


字符串题型

- ◎ 基于字符计数的问题
- ◎ 同字母异序
- ◎ 回文
- ◎ 二进制字符串
- ◎ 子序列
- ◎ 模式搜索
- ◎ 其他

子序列 I

◎ 最长子序列

- ◎ 给定一个字符串 's' 和一个整数k，找到其他字符串 't'，使得 't' 是给定字符串 's' 的最大子序列，同时 't' 的每个字符在字符串s中必须至少出现k次。
- ◎ Input: s = "baaabaacba "， k = 3
- ◎ Output : baaabaaba

子序列 II

◎ 检查子序列

- ◎ 给定两个字符串str1和str2。 确定str1是否是str2的子序列。 子序列是可以通过删除一些元素而不改变其余元素的顺序从另一个序列派生的序列。
- ◎ Input: str1 = "AXY", str2 = "ADXCPY"
- ◎ Output: True (str1 是 str2的子序列)
- ◎ Input: str1 = "AXY", str2 = "YADXCP"
- ◎ Output: False (str1 不是str2的子序列)

子序列 III

- ◎ 通过删除给定字符串的字符得到字典中最长的单词
- ◎ 给一个字典和一个字符串 'str'，找到字典中最长的字符串，它可以通过删除给定的 'str' 中的一些字符来形成。
- ◎ Input: dict = {"ale", "apple", "monkey", "plea"},
- ◎ str = "abpcplea"
- ◎ Output : apple

子序列 IV

- ◎ 找出所有子列元素之和的加和。
- ◎ 给定一系列n个整数.。找出所有子列元素之和的加和。

```
Input : arr[] = { 6, 8, 5 }
```

```
Output : 76
```

```
All subsequence sum are:
```

```
{ 6 }, sum = 6
```

```
{ 8 }, sum = 7
```

```
{ 5 }, sum = 6
```

```
{ 6, 8 }, sum = 14
```

```
{ 6, 5 }, sum = 11
```

```
{ 8, 5 }, sum = 13
```

```
{ 6, 8, 5 }, sum = 19
```

```
Total sum = 76.
```

字符串题型

- ⊙ 基于字符计数的问题
- ⊙ 同字母异序
- ⊙ 回文
- ⊙ 二进制字符串
- ⊙ 子序列
- ⊙ 模式搜索
- ⊙ 其他

模式搜索 I

- ◎ strStr

- ◎ 敏感词

- ◎ 给定的句子作为输入，用星号 ‘*’ 检查替换的单词。

- ◎ 用C替换所有出现的字符串AB

- ◎ 给定一个可能包含一个“AB”的字符串str。将str中的所有“AB”替换为“C”。

- ◎ 数出“1 (0+) 1”模式的发生次数

- ◎ 给定一个字母数字字符串，找出给定字符串中出现模式1 (0+) 1的次数。这里，(0+) 表示存在连续0的非空序列。

模式搜索 II

◎ 与通配符匹配的字符串

◎ 给定两个字符串，其中第一个字符串可以包含通配符字符，第二个字符串是普通字符串 编写一个函数，如果两个字符串匹配，则返回true。 第一个字符串中允许使用以下通配符。

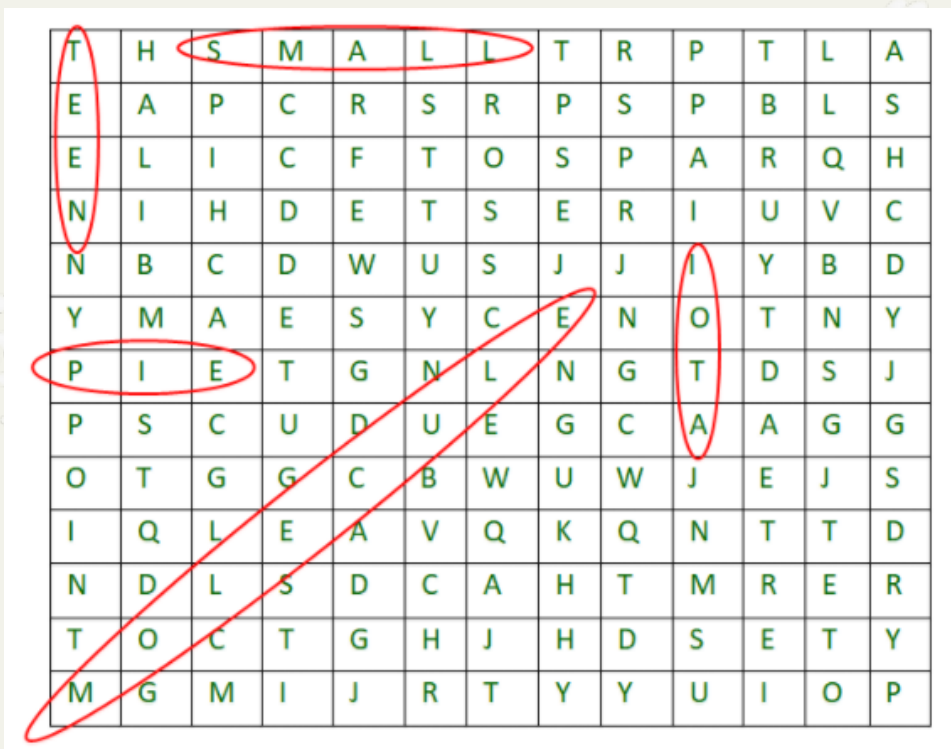
◎ * --> Matches with 0 or more instances of any character or set of characters.

◎ ? --> Matches with any one character.

模式搜索 III

◎ 二维网格单词搜索

- ◎ 给定一个词和一个二维字母网格，在网格中找到所有的给定词。每个点的八个方向均可进行匹配。若某方向上所有的字母均和单词匹配，则认为单词在此方向上被找到。
- ◎ 这八个方向分别是，直角方向和对角方向。



模式搜索 II

- 在矩阵中找到所有出现的给定词

- 给定一个二维网格的字符和单词，在网格中查找所有出现的给定单词。一个单词可以在一个点的8个方向上匹配。如果所有的字符都在这个方向上匹配（不是以之字形的形式），则可以说该单词被找到。
- 若出现循环则打印所有坐标。
- 这八个方向分别是，直角方向和对角方向。

	0	1	2	3	4
0	B	N	E	Y	S
1	H	E	D	E	S
2	S	G	N	D	E

	0	1	2	3	4
0	B	N	E	Y	S
1	H	E	D	E	S
2	S	G	N	D	E

数据结构与算法

Data Structure and Algorithm

XXIX 字符串

授课人: Kevin Feng

翻 译: 赵 其 辰