

算法分析与设计课程设计

Xiang Liu

2023-12-15

Contents

伪代码形式	1
1 课程设计要求	2
2 题组一	3
3 题组二	4
4 题组三	4
5 选作部分	6

伪代码形式

1. if 语句

```
if condition then
    statement1
else
    statement2
end
```

或者

```
if condition:
    statement1
else
    statement2
```

2. 循环语句

```
while condition do
    statement1
end
```

```
while condition:
    statement1
```

3. 函数

可以返回多个值，可以返回复杂结构体。

```
function fun(arg1, arg2)
    statement
    return one, two
end
```

```
function fun(arg1, arg2):
    statement
    return one, two
```

4. 数据结构

中括号表示线性表，大括号表示字典和集合结构。

```
list=[1,2,3]
set={1,2,3}
dict={"1":"hello",2:"world"}
```

在涉及到具体操作，例如删除，插入，查找，该操作可以用文字表述或者参照其他语言的习惯。

5. 注意事项

不要将伪代码写成详细代码，因为有时候文字表述比代码更清晰

1 课程设计要求

分组：每组由四名组员构成。组内成绩由学生自行评定，要求不同组员的成绩至少相差 5 分。每组至多允许一名申优者。

答辩时老师将随机抽取组内一人进行答辩，其他组员的成绩参考组内成绩差进行评分。

鼓励各个小组内部发挥互助友爱精神，促进充分讨论，力争每个人都能准确理解算法并亲自动手实现算法。

文档：每人提交一份文档，内容的要求如下：

1. **文字表达**: 请努力组织语言, 确保表述清晰明了
2. **代码编写**: 请使用清晰易懂的伪代码, 切勿粘贴真实代码, 更不能简单使用截图代替。
3. **动态规划**: 详细解释目标函数的递推公式, 能阐述问题规模变小的过程以及原问题如何依赖于子问题。请避免使用”状态转移方程”这个术语。
4. **回溯法或分支限界**: 在讨论时请注意状态空间树的建立和描述, 遵循状态空间树的深度优先遍历技术路线。

成绩评定:

1. 及格: 完成题组中指明必须完成的题目
2. 其他: 视完成情况和理解程度而定。

2 题组一

前四题必须完成, 而后三题可选, 第五题相对简单。

1. 最长回文子序列 (#516)

<https://leetcode.cn/problems/longest-palindromic-subsequence/>

2. 最长递增子序列 (#300)

<https://leetcode.cn/problems/longest-increasing-subsequence/>

3. 最长公共子序列 (#1143)

<https://leetcode.cn/problems/longest-common-subsequence/>

4. 编辑距离 (#72)

<https://leetcode.cn/problems/edit-distance/>

5. 最长等差子序列 (#1027)

<https://leetcode.cn/problems/longest-arithmetic-subsequence/>

6. 雀魂启动

<https://www.nowcoder.com/questionTerminal/448127caa21e462f9c9755589a8f2416>

7. 通配符匹配

<https://leetcode.cn/problems/wildcard-matching/>

3 题组二

前三题必须完成，而后三题中任意选择两题。

1. 为了让大家了解和学习并查集，请各位学习教材 9.2 节，然后利用并查集实现最小生成树的 Kruskal 算法。需要有最小生成树的直观界面展示（一般语言都有绘制网络图的现成工具包）。
2. 据说从湖北出发最多只需要跨越两个省就可以到达中国任何一个省。请设计实验对此进行验证。
3. 设计实验比较三分查找算法和二份查找算法平均情形时间复杂度。
4. 随机产生平面若干点，利用蛮力算法和分治算法找到平面的最接近点对，并考查随 n 变大时，两者的效率差异、实验效率和理论效率的一致性。平面点集能直观的进行观察。
5. 实现任务分配问题（随机产生不同的代价矩阵）的蛮力算法、迭代算法（匈牙利算法）、分支限界算法，并对后两者的效率进行比较和分析。
6. 随机产生 n 个顶点的有向图，并预设两点之间出现有向边的概率，然后探索前两个参数和图中出现环的关系。需要有图形展示。

4 题组三

至少选择 2 题完成。

1. 电话号码的字母组合

给定一个仅包含数字 2-9 的字符串，返回所有它能表示的字母组合。答案可以按任意顺序返回。给出数字到字母的映射如下（与电话按键相同）。注意 1 不对应任何字母。



Figure 1: 电话按键分布

示例 1:

输入: `digits = "23"`
输出: `["ad","ae","af","bd","be","bf","cd","ce","cf"]`

示例 2:

输入: digits = “”
输出: []

示例 3:

输入: digits = “2”
输出: [“a”, “b”, “c”]

提示:

$0 \leq \text{digits.length} \leq 4$
digits[i] 是范围 [‘2’, ‘9’] 的一个数字。

2. 括号生成

数字 n 代表生成括号的对数，请你设计一个函数，用于能够生成所有可能的并且有效的括号组合。

示例 1:

输入: n = 3
输出: [“((())”, “(())()”, “()()()”, “()()()”]

示例 2:

输入: n = 1
输出: [“()”]

提示:

$1 \leq n \leq 8$

3. 单词搜索

给定一个 $m \times n$ 二维字符网格 board 和一个字符串单词 word 。如果 word 存在于网格中，返回 true ；否则，返回 false 。

单词必须按照字母顺序，通过相邻的单元格内的字母构成，其中”相邻” 单元格是那些水平相邻或垂直相邻的单元格。同一个单元格内的字母不允许被重复使用。

示例 1:

A	B	C	E
S	F	C	S
A	D	E	E

输入: board = [[“A”, “B”, “C”, “E”], [“S”, “F”, “C”, “S”], [“A”, “D”, “E”, “E”]],
word = “ABCCED”
输出: true

示例 2:

A	B	C	E
S	F	C	S
A	D	E	E

输入: board = [[“A”, “B”, “C”, “E”], [“S”, “F”, “C”, “S”], [“A”, “D”, “E”, “E”]],
word = “SEE”
输出: true

示例 3:

A	B	C	E
S	F	C	S
A	D	E	E

输入: board = [[“A”, “B”, “C”, “E”], [“S”, “F”, “C”, “S”], [“A”, “D”, “E”, “E”]],
word = “ABCB”
输出: false

提示:

m == board.length
n = board[i].length
1 <= m, n <= 6
1 <= word.length <= 15
board 和 word 仅由大小写英文字母组成

进阶: 你可以使用搜索剪枝的技术来优化解决方案, 使其在 board 更大的情况下可以更快解决问题?

5 选作部分

1. 迷宫寻路问题。

1. 产生图形用户界面, 设计一个迷宫生成算法, 使得产生的每个迷宫都是有解的
2. 能绘制出从起点到终点的路径。路径通过 BFS 和 DFS 遍历得到。
3. 实现 A* 算法, 将 A* 算法的运行和 BFS 的运行进行比较 (产生不同的迷宫, 分别运行 A* 和 BFS 算法, 多次比较他们的快慢)

参考:

- i. <https://www.bilibili.com/video/BV1eX4y1v7rS/>
- ii. <https://www.youtube.com/watch?v=-L-WgKMFuhE>

2. 文章查重系统

班级抄袭之风盛行，为了便于检测学生提交的课程报告之间的相似度，皇家理工 AI 学院决定让学生开发一个查重系统。

输入：一个指定目录。其中包含数十份学生的课程设计报告。

输出：提供每份课程设计报告的抄袭率。

算法：

1. 两个段落之间的抄袭长度

首先，确定两个段落之间被视为抄袭的最小文本长度。例如，您可以定义从另一个段落复制的连续包含 10 个单词（汉字）的段将构成抄袭。

然后，逐字比较段落，以查找符合或超过定义的抄袭长度的任何匹配片段。跟踪找到的最长匹配字符串中的单词数。如果匹配段大于或等于定义的抄袭长度，则两个段落之间发生抄袭。抄袭的程度可以通过复制片段的长度之和来量化。

两个段落 p_1, p_2 之间的抄袭长度定义为：

$$\text{dup_p2p}(p_1, p_2)$$

其中， p 表示段落（paragraph）。

2. 段落和文章之间的抄袭长度

定义段落和文章的抄袭长度 $\text{dup_p2a}(p, a)$ 为段落和另一篇文章所有段落之间的最大抄袭长度：

$$\text{dup_p2a}(p, a) := \max_{q \in a} \text{dup_p2p}(p, q)$$

a 表示文章（article）。

3. 文章之间的抄袭长度

$$\text{dup_a2a}(a, b) := \sum_{p \in a} \text{dup_p2a}(p, b)$$

4. 文章的抄袭率

一篇文章的抄袭率是这篇文章和其他所有文章比对后的最大抄袭长度除以该篇文章的长度。

$$\text{dup}(a) := \max_{b \text{ in all articles}} \text{dup_a2a}(a, b) / \text{Total length of the article}$$

请问，以上定义是否存在不合理处？应该如何改进。

任务：

1. **制作或者下载查重文章集**：可通过文本生成方法，如马尔科夫链，制作或下载一系列具有差异性的文章，以测试查重系统的性能。

2. **抄袭率计算与匹配文章**：开发的系统能计算出每篇文章的抄袭率，同时指出与之匹配的可能抄袭文章。
 3. **其他抄袭率算法设计**：尝试设计不同的抄袭率算法，可以考虑使用哈希函数、词向量模型等方法，以提高查重系统的多样性和准确性。
3. 学校超市选址问题（带权有向图的中心点）
- 设计内容：对于某一学校超市，其他各单位到其的距离不同，同时各单位人员去超市的频度也不同。请为超市选址，要求实现总体最优。
- 设计要求：
- (1) 设计该问题的核心算法；
 - (2) 设计可视化的界面，界面中能有效显示学校超市可设立的地点和各单位的位置以及它们之间的有效路径；
 - (3) 程序能自动计算出最优设立点，并以图示化方式演示。