

算法设计与分析——作业四

作业提交截止日期：2025.6.17 23:59:59

提交文件列表如下：

文件内容	文件格式	文件名
书面作业	.pdf	学号-姓名-hw4-书面作业.pdf
编程题的代码	/	学号-姓名-hw4-编程代码
编程题的文档	.pdf	学号-姓名-hw4-编程文档.pdf

① Note

将以上内容打包为一个.zip文件，命名为“学号-姓名-hw4-x.zip”，其中x是你所在班级的编号，具体如下：

罗烨老师上午班级：1，罗烨老师下午班级：2，朱亚萍老师班级：3，张苗苗老师班级：4

发送邮箱：2433285@tongji.edu.cn，邮件的主题即为“学号-姓名-hw4-x”，上传作业若成功，会收到回复邮件。

⚠ Caution

注意：提交作业的邮箱与第二次作业相同；编程题请使用华为云平台完成，要求同作业三。

作业要求：

1. 请尽量使用电子版的方式完成答题，如提交手写版的作业，请保持书写端正并将其扫描为足够清晰的pdf文件
2. 解答题的部分需要写明解题过程，若只给答案会扣除一定分数
3. 编程题目文档需要包括算法实现核心代码及其解释，测试用例通过的截图（截图中需要包含输入和输出）。缺少通过截图不得分

4. ⚠ Warning

切勿抄袭（抄袭同学或抄袭网上答案），如发现抄袭的情况，本次作业（抄袭者与被抄袭者）按照零分处理

一、概念梳理题（20%）

1. 动态规划算法是一种求解多阶段决策过程最优化问题的方法。其核心思想是：把原问题分解为若干个重叠的_____，每个子问题的求解过程都构成一个阶段。在完成一个阶段的计算之后，动态规划方法执行下一个阶段的计算。
2. 在求解子问题的过程中，按照_____（自顶向下/自底向上）的记忆化_____方法或者_____（自顶向下/自底向上）的_____方法求解出_____，把结果存储在表格中。当需要再次求解此子问题时，直接从表格中_____即可，从而避免了大量的_____。
3. 能够使用动态规划解决问题必须满足以下三个特征：①_____，②_____，③_____。
4. 动态规划的关键是_____，即如何由已求出的_____来推导全局最优解。

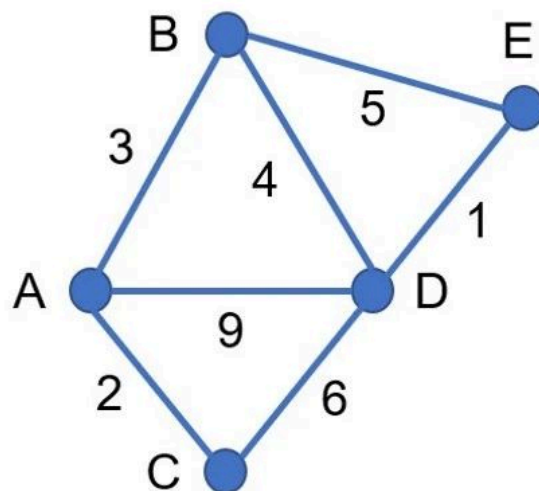
5. 动态规划算法中，最简单的，可以直接得出的局部最优解，被称为_____。
6. 贪心算法一步一步地进行_____，希望通过一系列的选择得到_____。
7. 贪心选择性质是指，所求问题的整体最优解可以通过一系列_____，即贪心选择，来达到。这是贪心算法可行的第一个基本要素，也是贪心算法与动态规划算法的主要区别。
8. 若一个优化问题的优化解包含它的_____的优化解，则称其具有_____。
9. 通过局部最优解的计算，贪心算法_____（一定/不一定）能得到全局最优解。

二、解答题 (40%)

1. 一层楼有8级台阶，一个人要上一层楼，他可以一步走1级台阶或2级台阶。请问他有多少种上楼的方法？请你使用动态规划的思想解决这个问题，你需要解释如何在本题中使用动态规划的思想，并写出动态转移方程和边界条件，并最后逐步得出答案。 (7%)

2. 完成下列问题 (4%+7%)

- (1) 简述迪杰斯特拉 (Dijkstra) 算法的基本思想和基本流程，其体现的是什么算法的思想？
- (2) 对于下图，假定从节点A出发，使用迪杰斯特拉 (Dijkstra) 算法寻求到达节点E的最短路径。



3. 在平时的生活中，若使用现金支付，我们往往会遇到现金找零的问题，我们总是希望能使用最少的现金张数完成找零。在这其中就蕴含着贪心算法的思想。假设纸币的面额所构成的集合为 M 。请你完成下列问题（2%+5%+2%+3%+8%+2%）

（1）以我们平时生活中使用的面额为例，即 $M = \{1, 5, 10, 20, 50, 100\}$ ，依据生活常识，我们知道这种情况下满足贪心选择性质。若需要找零67元，模拟贪心算法的步骤，写出找零的过程。若使用贪心算法，需要找几张现金？

（2）若 $M = \{1, 2, 4, 8, 16\}$ ，是否还满足贪心选择性质？说明原因。

（3）根据第（2）问中的集合 M 的选择，提出一个更一般化的猜想。（无需证明）

（4）是否存在不满足贪心选择性质的集合 M ？若有，请举出一个例子，并说明它为什么不满足贪心选择性质。

（5）为了凑出 n 元找零，对于一个含有 k 个元素的集合 M （为了确保能找出零钱，我们默认 $1 \in M$ ）。为了确保找零时总能使用最少的现金张数，设计一个动态规划的找零算法，并求其时间复杂度。（无需写代码，只需文字说明即可，可以使用数学公式辅佐你的说明）

（6）通过本题可以得知，贪心算法和动态规划算法最终得出的结果有什么区别？

三、编程题（40%）

在作业二中，我们学习到了有关图像存储的方法。这里再次复习一下：在图像存储的时候，我们可以使用不同的数值对应不同像素点的颜色。例如，若一张图像只采用黑，白二色进行存储，我们可以用0表示黑色像素点，用1表示白色像素点。

现考虑一张大小为 $M \times N$ 的矩形的黑白图像。请求出图像中最大的白色矩形的面积。

输入第一行： M和N的大小

输入第二行及以后： 图像所对应的像素点矩阵

输出： 图像中最大的白色矩形的面积

示例1

输入

```
2 3
0 1 0
0 0 0
```

输出

```
3
```

示例2

输入

```
4 5
1 0 0 1 0
1 0 0 0 0
1 0 0 0 0
1 0 1 1 0
```

输出

```
8
```

完成下列习题

Tip

以下内容请写在编程题文档中

- (1) 该问题可以采用哪种解决问题的策略解决？结合相关概念进行分析。
- (2) 使用C++语言编写相关代码，并在文档中给出算法实现核心代码及其解释
- (3) 在文档中给出测试用例通过的截图（截图中需要包含输入和输出，可以适当加入一些输入输出提示语句）

【测试用例截图共需三个：① 示例1，② 示例2，③ 自己编写一个测试用例】

- (4) 在利用算法求解示例2时，模拟算法的求解过程，写出算法执行过程中，中间计算生成的表格。并大致解释该表格的含义。

- (5) 分析你的代码中的函数的时间复杂度