**算法设计与分析第四章作业**

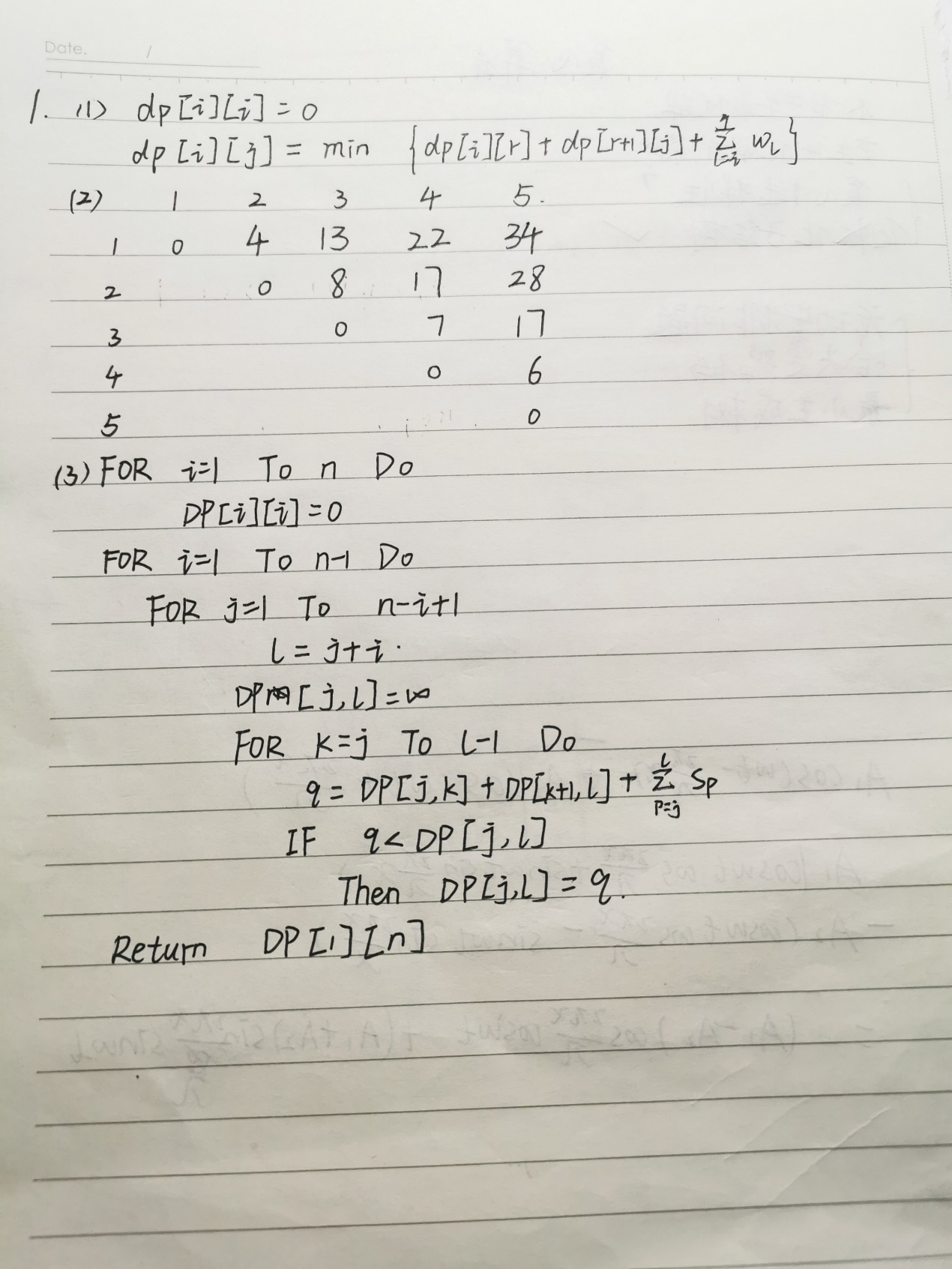
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** |  | **班级** |  | **学号** |  |
| **第1题** |  | | | | |
| **第2题** |  | | | | |
| **第3题** |  | | | | |
| **第4题** |  | | | | |
| **第5题** |  | | | | |
| **第6题** |  | | | | |
| **总分** |  | | | | |
| **备注** | 作业提交截止时间：2020-10-14日24:00，超过提交截至时间的作业视为无效。确因网络等特殊原因无法及时提交作业的学生，应至少提前1小时与助教。作业文件名命名方式： 第x章-x班-姓名-学号（例， 第1章-1班-张三-2018054000.doc）； 邮件主题为：第x章作业, x班，姓名，学号（例， 第1章作业，1班，张三，2018054000）。缺少这些信息的作业将被酌情扣分。 | | | | |

1. （30分）在一条直线上有n堆石子，每堆有一定的数量，每次可以将两堆相邻的石子合并，合并后放在两堆的中间位置，合并的费用为两堆石子的总数。求把所有石子合并成一堆的最小花费(定义dp[i][j]为第i堆石子到第j堆合并的最小花费)。

(1)写出该问题的递推方程。（10分）

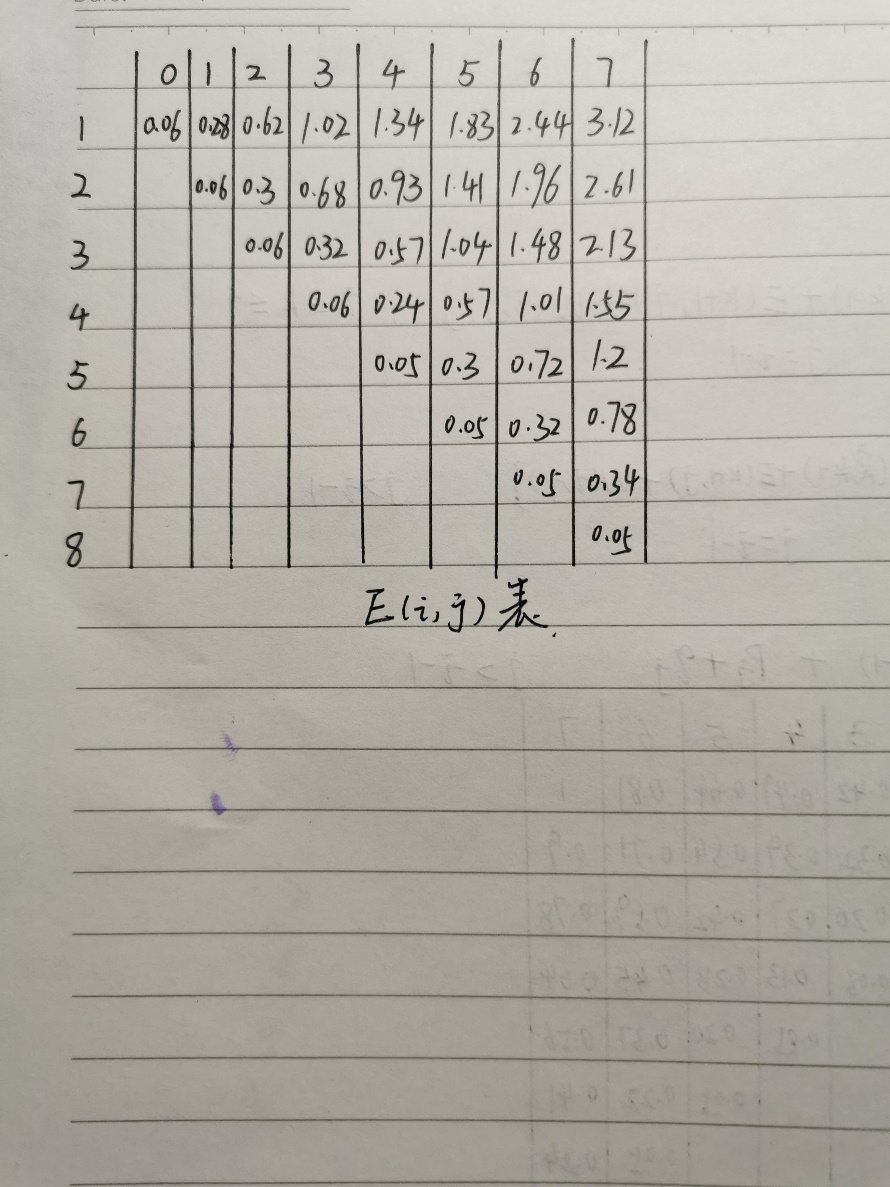
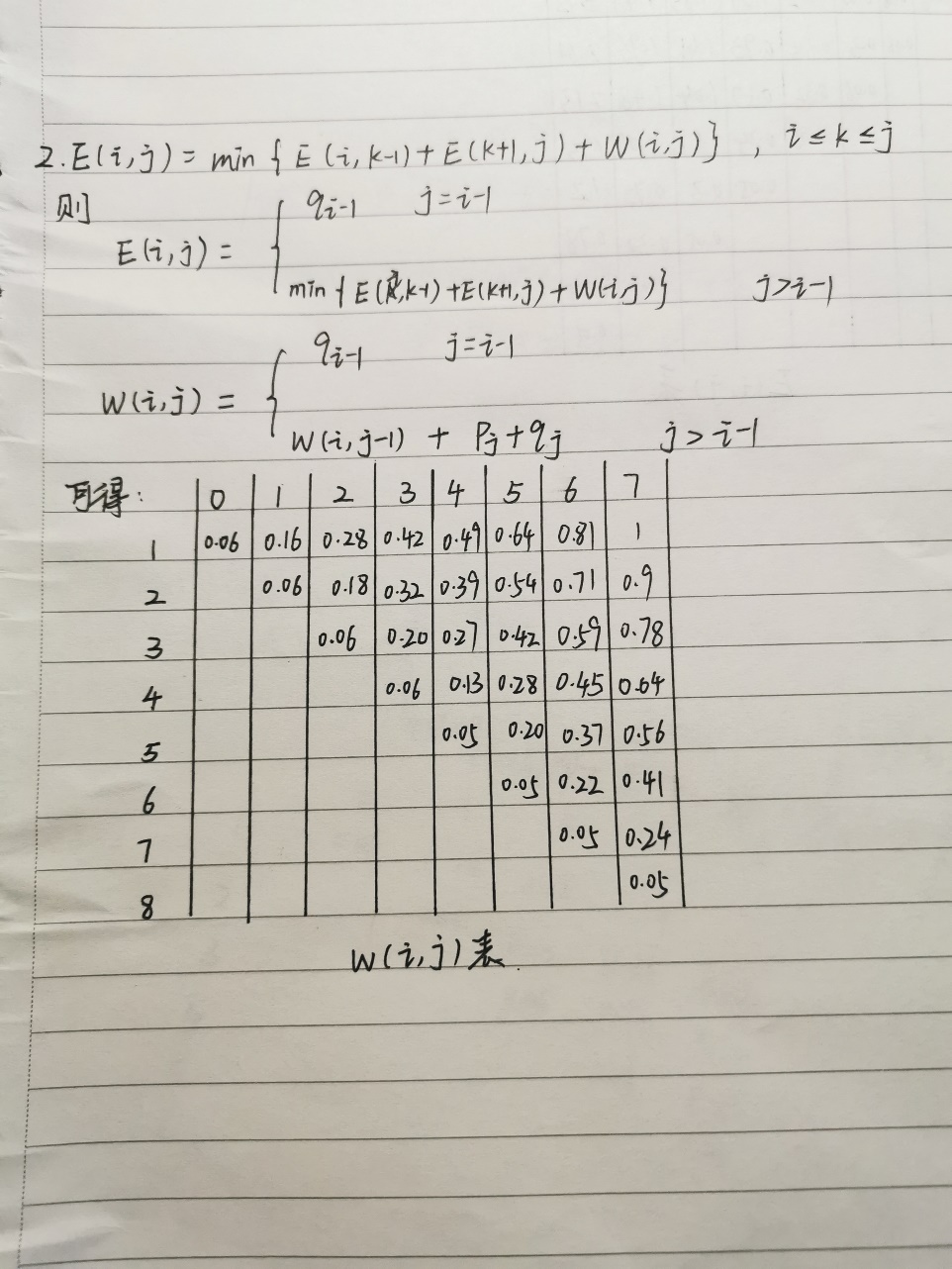
(2)有5堆石子(n=5)，每堆石子大小分别为<1,3,5,2,4>,求出把所有石子合并成一堆的最小花费(要求写出运算矩阵)。（10分）

(3)写出该问题的伪代码。（10分）



1. 若7个关键字的概率如下所示，求其最优二叉搜索树的结构和代价，要求必须写出递推方程。（30分）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  | 0.04 | 0.06 | 0.08 | 0.02 | 0.10 | 0.12 | 0.14 |
|  | 0.06 | 0.06 | 0.06 | 0.06 | 0.05 | 0.05 | 0.05 | 0.05 |



1. 编程题：兑换零钱问题（40分）

题目描述：

给定不同面额的硬币 coins 和一个总金额 amount。编写一个函数来计算可以凑成总金额所需的最少的硬币个数。如果没有任何一种硬币组合能组成总金额，返回-1。（提示：你可以认为每种硬币的数量是无限的）。

示例 1:

输入: coins = [1, 2, 5], amount = 11

输出: 3

解释: 11 = 5 + 5 + 1

示例 2:

输入: coins = [2], amount = 3

输出: -1

**要求：**

**运用动态规划的思想作答，请写出分析过程和状态转移方程，并用一种语言（最好是C++或JAVA）实现你的思路（粘贴在交作业的那个word里），并保证代码能正确运行，复杂度尽可能低。**

System.***out***.println(**"请输入coin数组:"**);  
Scanner sc = **new** Scanner(System.***in***);  
ArrayList<Integer> list = **new** ArrayList();  
**int** count = 0;  
**while**(!sc.hasNext(**"#"**)){  
 list.add(sc.nextInt());  
 count++;  
}  
String str = sc.next();  
System.***out***.println(**"请输入amount"**);  
**int** amount = sc.nextInt();  
**int** arr[] = **new int**[amount+1];  
arr[0] = 0;**for** (**int** i = 1; i <= amount; i++) {  
 arr[i] = -1;  
 **for**(**int** j = 0; j < count; j++){  
 **if**(i-list.get(j) >= 0)  
 {  
 **int** temp = arr[i-list.get(j)];  
 **if**(temp >= 0){  
 **if**(arr[i] == -1 || temp+1 <arr[i]) arr[i] = temp+1;  
 }  
 }  
 }  
}  
return arr[amount];

