**集美大学实验报告**

**班级 计算2114 学号 202121331104 姓名 庄佳强 成绩**

**课程 《算法设计与分析》 日期 2024 年 3 月 26 日**

|  |  |  |  |
| --- | --- | --- | --- |
| **实验项目** | **蛮力法应用** | **指导老师** | **杨艳华** |
| **实验目的** | 掌握蛮力法的使用 | | |
| **一、实验内容**  1.某个国家仅有1分、2分和5分硬币，将钱n(n≥5)兑换成硬币有很多种兑法。编写一个程序计算出给定的钱数有多少种兑法，并列出每种兑换方式。  2.假设有n个任务需要分配给n个人执行，每个任务只分配给一个人，每个人只执行一个任务，且第i个人执行第j个任务的成本是Cij（1≤i, j≤n），任务分配问题要求找出总成本最小的分配方案。 | | | |
| **二、问题分析**  1.第一题中可以直接使用循环枚举出三种硬币的使用可能性，然后相加为n的保留输出。  2.第二题可以使用全排列找出所有人的所有不同的排列方式。然后查看当前组合的成本是否比之前的成本更新，是否需要更新最小成本。 | | | |
| **三、数据结构定义**  1.第二题中，定义一个二维数组用来存储输入的数值，定义了一个一维数组用来存放最小结果排列。 | | | |
| **四、算法伪代码描述**   1. 第一题：   function fun(n)          sum<-0        for i<-0 toi<n/5 by i++ do          for j<-0 j<=(n-i\*5)/2 by j++ do            for k<-0 k<=n-i\*5+j\*2 by k++ do              if i\*5 + j \* 2 + k  == n  sum++;  printf("%03d%12d%12d%12d\n", sum, k, j, i);             end if      end for     end for   end for  end function   1. 第二题：   function solution(int[][]arr)     size<-arr.size();  res <-INT.MAX  int resarr[]     for i<-0 i<size by i++ do     b[i]=i  end for  do   for i<-0 i<size by i++ do     min+=arr[i][b[i]]   end for   if(min<res)    res=min      对resarr进行赋值   end for    for i<-0 i<size by i++ do       打印结果  end function | | | |
| **五、算法时间和空间复杂度分析（要求写出详细过程）**  第一题：  因为使用了三层循环：  if i\*5 + j \* 2 + k  == n指令要执行约(n/5)((∑(n-i))\*5/2)(n)次  T(n)=O(n^3)  因为没有使用额外的空间：空间复杂度为：O(1)。  第二题：  主要的二重循环中，一个for循环进行了n次，而外围的全排列函数会执行n！次，所有T(n)=O(n!\*n)  题目中使用了一个二维数组存储输入内容，因而空间复杂度为O(n^2) | | | |
| **六、错误记录和解决**  在第二题中，写完代码发现只能过一个检测点，但是代码完全没有问题，在本次测试了多个测试都没什么问题，一是不知道怎么办。  解决：在同学的提醒下，得知如果同最小值，本题中要取后手，在判断最小值中加上等号就解决了。 | | | |
| **七、心得体会**  通过本次实验，我学到了蛮力法在解决问题中的可能性，虽然效率很低下，但能解决问题，在后续中可以使用更好的算法进行替代。 | | | |