**《算法设计与分析》综合实验**

1. **实验题目**

1.请用两种以上的算法解决下面问题，并分析不同算法的时间复杂度和各自的优劣。

案例描述：

爸爸规划一家人暑假出去自驾游，假设爸爸爱车的油箱加满油后可行驶M公里，自驾游的路线沿途有n个加油站，加油站之间的距离由数组D[n]给出，其中D[i]表示第i-1个加油站到第i个加油站之间的距离，假定旅途的起点和终点都各有一个加油站。设计一个算法，使得加油的次数最少。

输入：自驾游总路线长度：K,

          加满油可行驶公里数：M,

          沿途加油站数：n

          各加油站之间的距离数：D[n]

输出：最少加油次数c， 加油站停靠情况：（如： 1 0 0 1 1 0 0 0 1 0 。。。。。）

要求：

（1）完整代码；

（2）程序运行截图；

（3）实验报告。

1. **综合实验的报告内容如下：**

1. 问题描述：描述要求解决的问题。

2. 按要求给出算法解决方案。

3. 详细描述所设计算法的设计思想。

4. 详述描述问题的解决过程。

5. 源程序：给出所有源程序清单，要求程序有充分的注释语句，至少要注释每个函数参数

6. 给利用一组实验数据进行算法程序验证并给出算法的执行结果。（可拷贝屏幕进行粘贴）

7. 给出本课程的学习收获和感想

1. **格式要求**
2. 题目用黑体三号，段后距18磅（或1行），居中对齐；
3. 标题用黑体四号，段前、段后距6磅（或0.3行）；
4. 正文用小四号宋体，行距为1.25倍行距；
5. 标题按“一”、“㈠”、“1”、“⑴”顺序编号。
6. 封面版式参考附件1。
7. **实验作业提交**

大作业上传至学习通平台。包括大作业报告和源程序代码。

**重庆交通大学2022-2023学年第一学期**

**《算法设计与分析》实验报告**

**题目：** 同一问题的不同解决方法设计与实现

学 院： 信息科学与工程学院

专 业： 计算机类

学 生 姓 名： 田雨

学 号： 632007060108

完 成 时 间： 2023年1月2日

重庆交通大学

CHONGQING JIAOTONG UNIVERSITY

一 问题描述

请用两种以上的算法解决下面问题，并分析不同算法的时间复杂度和各自的优劣。

案例描述：

爸爸规划一家人暑假出去自驾游，假设爸爸爱车的油箱加满油后可行驶M公里，自驾游的路线沿途有n个加油站，加油站之间的距离由数组D[n]给出，其中D[i]表示第i-1个加油站到第i个加油站之间的距离，假定旅途的起点和终点都各有一个加油站。设计一个算法，使得加油的次数最少。

输入：自驾游总路线长度：K,

          加满油可行驶公里数：M,

          沿途加油站数：n

          各加油站之间的距离数：D[n]

输出：最少加油次数c， 加油站停靠情况：（如： 1 0 0 1 1 0 0 0 1 0 。。。。。）

要求：

（1）完整代码

（2）程序运行截图

（3）实验报告

二 贪心算法方案

1. 贪心算法设计思想

通过贪心算法得到局部最优解，从问题的某一个初始解出发逐步逼近给定的目标，以尽可能快的地求得更好的解。当达到算法中的某一步不能再继续前进时，算法停止。每到达一个加油站，都要判断是否能到达下一个加油站，若不能，则在该站点加油，若能，不在该站点加油，继续行驶,该算法遍历所有的加油站，且不需要重复遍历，所以时间复杂度为O(n)。

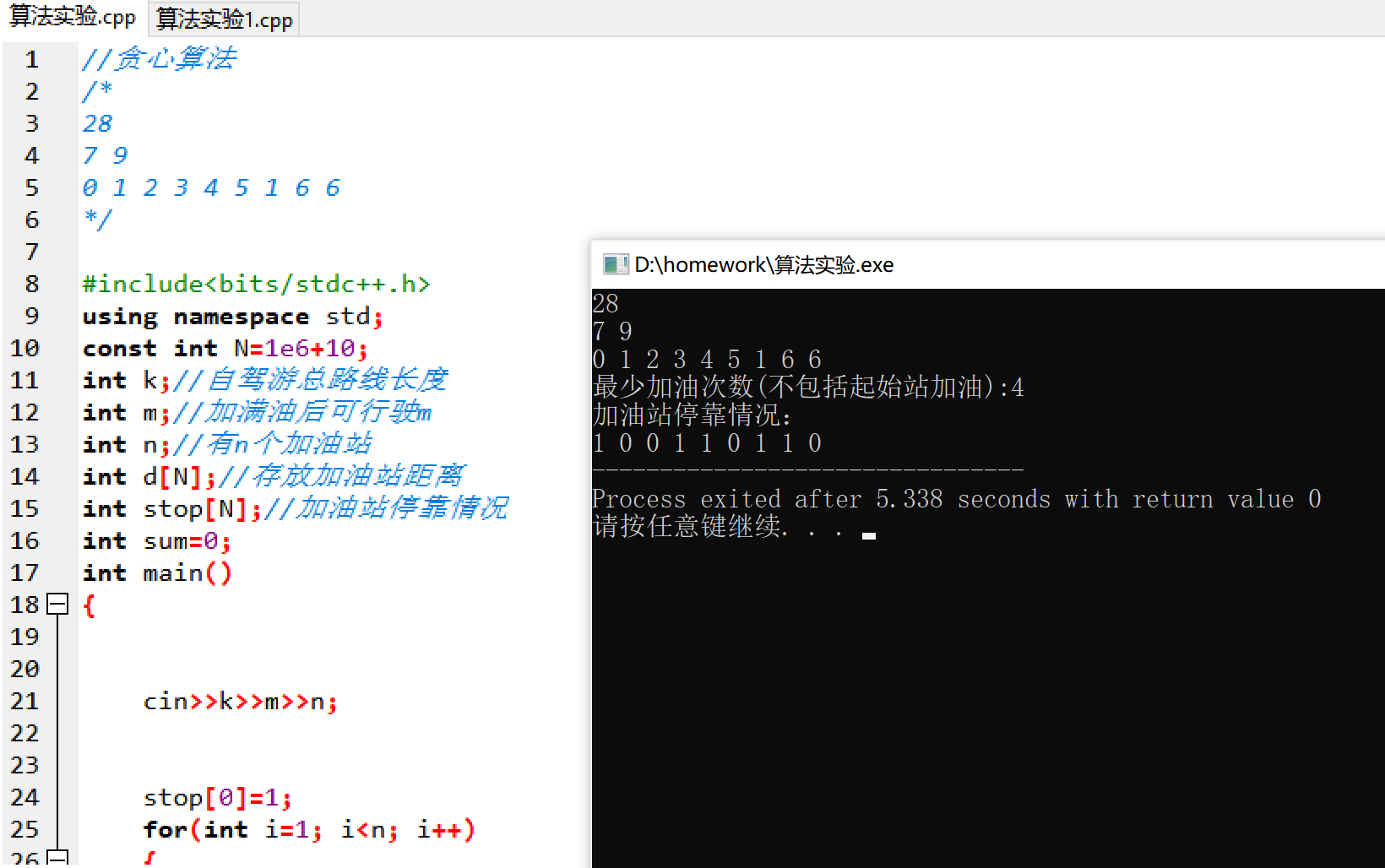
1. 贪心算法解决过程

假定汽车在起始站加满油后出发，即默认起始站加油，终点站不加油，之后每到达一个加油站，都要判断是否能到达下一个加油站，若不能，则在该站点加油，若能，不在该站点加油，继续行驶，每经过一个站点后都要减去消耗的油量

1. 贪心算法源程序

|  |
| --- |
| //贪心算法  /\*  28  7 9  0 1 2 3 4 5 1 6 6  \*/  #include<bits/stdc++.h>  **using** **namespace** std**;**  const int N**=**1e6**+**10**;**  int k**;**//自驾游总路线长度  int m**;**//加满油后可行驶m  int n**;**//有n个加油站  int d**[**N**];**//存放加油站距离  int stop**[**N**];**//加油站停靠情况  int sum**=**0**;**  int main**()**  **{**  cin**>>**k**>>**m**>>**n**;**    stop**[**0**]=**1**;**  **for(**int i**=**1**;** i**<**n**;** i**++)**  **{**  stop**[**i**]=**0**;**  **}**  **for(**int i**=**0**;** i**<**n**;** i**++)**  **{**  cin**>>**d**[**i**];**  **}**    int sum**=**0**;**  int gas**;**//gas表示当前的油还可以走多少km  gas**=**m**;**//开始汽车加满油    **for(**int i**=**0**;** i**<**n**;** i**++)**  **{**  **if(**m**<**d**[**i**])**  **{**  cout**<<**"无法到达终点"**<<**endl**;**  **return** 0**;**  **}**  **if(**gas**<**d**[**i**])** //无法到达下个加油站  **{**  sum**++;**  stop**[**i**-**1**]=**1**;**  gas**=**m**;**//加油  **}**  gas**=**gas**-**d**[**i**];**//减去耗费的油量  **}**  cout**<<**"最少加油次数(不包括起始站加油):"**<<**sum**<<**endl**;**  cout**<<**"加油站停靠情况："**<<**endl**;**  **for(**int i**=**0**;** i**<**n**;** i**++)**  **{**  cout**<<**stop**[**i**]<<**" "**;**  **}**  **return** 0**;**  **}** |

1. 贪心算法执行结果



三 递归算法方案

（一）递归算法设计思想

递归算法，就是程序中不断反复的调用自身来解决问题的方法。这里强调的重点是调用自身，就是要求解的问题能够被分解成多个相同的小问题这样通过多次递归调用，完成求解。

递归算法的具体实现过程一般通过函数来完成。在函数的内部，编写程序时，直接或者间接的调用函数自己，即可完成递归操作。这种函数也可以叫做递归函数。在递归函数中，主调函数同时也时被调函数。执行递归函数将反复的调用自身，每调用一次就进入新的一层。

（二）递归算法解决过程

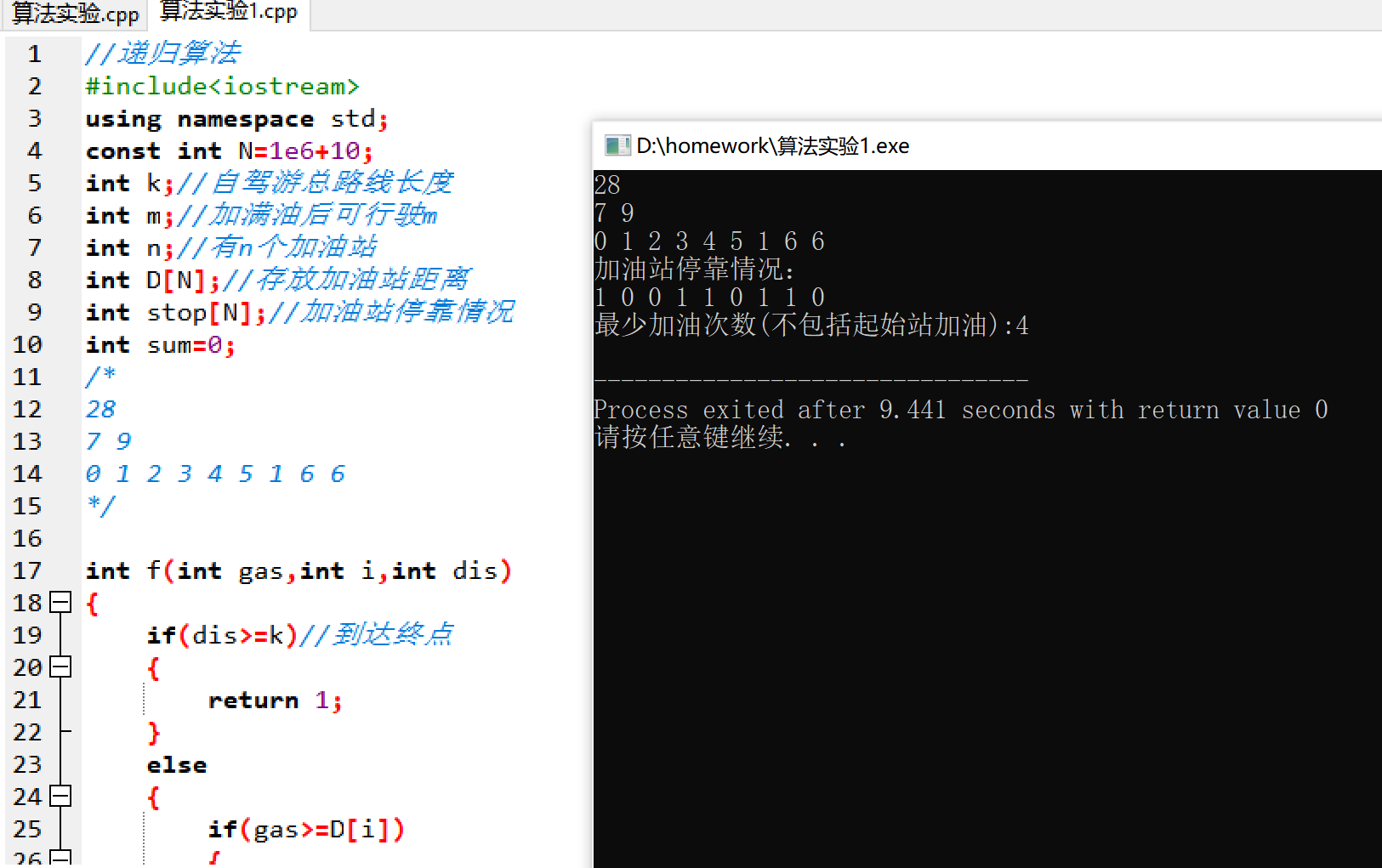
假定汽车在起始站加满油后出发，即默认起始站加油，初始行驶距离是0，

然后进入递归函数，进行判断，如果能到达下个加油站，将行驶距离和油量处理后再次递归，看否能到达下一个加油站，若不能，则在该站点加油后再递归，若能，不在该站点加油，继续行驶，每经过一个站点后都要减去消耗的油量

（三）递归算法源程序

|  |
| --- |
| //递归算法  #include<iostream>  **using** **namespace** std**;**  const int N**=**1e6**+**10**;**  int k**;**//自驾游总路线长度  int m**;**//加满油后可行驶m  int n**;**//有n个加油站  int D**[**N**];**//存放加油站距离  int stop**[**N**];**//加油站停靠情况  int sum**=**0**;**  /\*  28  7 9  0 1 2 3 4 5 1 6 6  \*/  int f**(**int gas**,**int i**,**int dis**)**  **{**  **if(**dis**>=**k**)**//到达终点  **{**  **return** 1**;**  **}**  **else**  **{**  **if(**gas**>=**D**[**i**])**  **{**  int x**;**  dis**=**dis**+**D**[**i**];**  gas**=**gas**-**D**[**i**];**  x**=**f**(**gas**,**i**+**1**,**dis**);**//看能否到达下个加油站  **if(**x**==**0**)**//不能到达，加油并记录次数  **{**  stop**[**i**]=**1**;**  sum**++;**  gas**=**m**;**  f**(**gas**,**i**+**1**,**dis**);**  **}**  **}**  **else**//无法到达  **{**  **return** 0**;**  **}**    **}**  **}**  int main**()**  **{**  cin**>>**k**>>**m**>>**n**;**  **for(**int i**=**0**;**i**<**n**;**i**++)**  **{**  cin**>>**D**[**i**];**  **}**    stop**[**0**]=**1**;**  int gas**=**m**;**//起始站加油  int dis**=**0**;**//记录汽车开过的距离    **if(**f**(**gas**,**1**,**dis**))**  **{**  cout**<<**"加油站停靠情况："**<<**endl**;**  **for(**int i**=**0**;**i**<**n**;**i**++)**  **{**  cout**<<**stop**[**i**]<<**" "**;**  **}**  **}**  **else**  **{**  cout**<<**"无法到达终点"**<<**endl**;**  **}**  cout**<<**endl**;**  cout**<<**"最少加油次数(不包括起始站加油):"**<<**sum**<<**endl**;**  **return** 0**;**  **}** |

（四）递归算法执行结果



四 收获和感想

通过这学期对本课程的学习，本人了解到了许多不同的算法，例如贪心算法、递归算法、动态规划、蛮力法、回溯算法等等，每种算法都有与之相对应的应用场景。同时，通过课堂上的练习以及课下的实验，提高了我们分析问题和解决问题的能力，让我学会了如何运用算法来解决实际的问题，这门课的学习让我收获颇丰，我也将继续学习并提升自己。