车厢调度问题实习报告

班级：电信1809 姓名：张浩 学号：U201813496

题目： 假设停在铁路调度站入口处的车厢序列的编号为1,2,3,4…n。设计一个程序，求出所有可能由此输出的长度为n的车厢序列。

1. 需求分析：

演示程序以用户和计算机的对话方式执行，即在计算机终端上显示“提示信息”之后，由用户在键盘上输入演示程序中规定的运算命令，相应地输入数据，运算结果显示在其后。

1. 概要设计：

为实现上述功能，选择栈作为抽象数据结构，车厢调度问题抽象为元素的一系列对栈的操作。

1. 详细设计：

#include <stdio.h>

#include <stdlib.h>

#define stacksize 100 //定义栈的深度

int n,num=0; //n表示待处理的车厢数，

num表示总可能数

typedef struct stack

{

int top; //定义一个栈类型

int Arr[stacksize];

}Stack,\*pStack;

Stack S;

void Init() //以下为对栈的一系列操作

{

S.top=-1;

}

void push(int e)

{

S.top++;

S.Arr[S.top]=e;

}

void pop()

{

S.Arr[S.top]=0;

S.top--;

}

int empty()

{

if(S.top==-1) return 1;

else return 0;

}

int Gettop()

{

if(S.top!=-1)

return S.Arr[S.top];

else return 0;

}

void possible(int pos,int Order[],int step) //输出所有可能

情况（递归）

{

if(pos <= n)

{

push(pos);

possible(pos+1,Order,step);

pop();

}

if(!empty())

{

int i=Gettop();

pop();

Order[step]=i;

possible(pos,Order,step+1);

push(i);

}

if(pos>n&&empty()==1)

{

for(int k=0;k<n;k++)

{

printf("%4d",Order[k]);

}

printf("\n");

num++;

}

}

void main()

{

printf("Input the number of trains");

scanf("%d",&n);

int Order[stacksize];

int cnt=0;

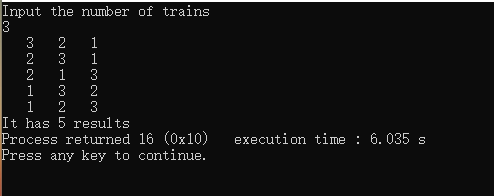
Init();

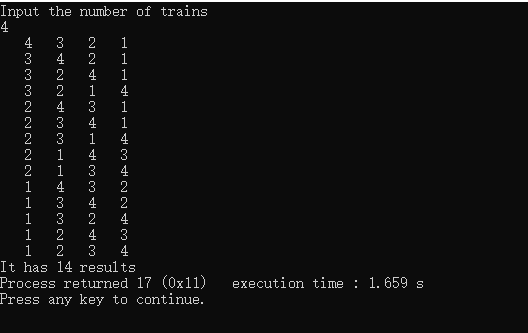
possible(1,Order,0); //从第一个开始处理

printf("It has %d results",num);

}

1. 调试分析：
2. 由于对递归过程的不了解，导致程序运行时出现的错误难以有效分析。
3. 刚开始用栈指针作为形参，发现调试出现了问题，最终采取在全局定义一个栈的方法。
4. 时间复杂度分析：在possible函数中，有两重递归，整个程序的时间复杂度是O(n²)。
5. 调试结果；





分别以n=3和n=4为例，调试程序，结果均正确。