实验报告

题目：设计程序利用栈的顺序存储结构解决车厢调度问题

班级：电信1809班 姓名：程铖

学号：U201813502 完成日期：2019/4/13

1. 需求分析
2. 输入车厢总数n；
3. 求出所有可能由此输出的长度为n的车厢序列；
4. 测试数据：取n=3，输出结果为321，231，213，123
5. 概要设计

利用栈的顺序存储结构以及利用递归来实现。

设计思想：利用栈的特性，一个数进栈以后，要么立刻出栈要么立刻下一个数进栈，出栈以后，也有两种处理方式：要么立刻出栈。要么下一个数进栈。以此即可用两重递归达到解决问题的目的。

程序主要包含四个模块

1. 主函数模块；
2. 栈顶元素的出栈；
3. 元素的入栈；
4. 检查栈是否为空；
5. 详细设计

时空分析：利用栈的特性结合递归使得能够利用较短的代码从而达到目的，push和pop函数实现出栈和入栈操作。

#include "pch.h"

#include <iostream>

#define MAX 100

int n;

typedef struct

{

int \*base;

int \*top;

int stacksize;

int data[MAX];

int m;

}SqStack;

void InitStack(SqStack &s)

{

s.base = (int \*)malloc(n \* sizeof(int));

if (!s.base) exit(0);

s.top = s.base;

s.stacksize = n;

s.m = -1;

}//初始化

void Push(SqStack &s, int e)

{

if (s.top - s.base >= n)

{

s.base = (int \*)realloc(s.base, (s.stacksize + 1) \* sizeof(int));

if (!s.base) exit(0);

s.top = s.base + s.stacksize;

s.stacksize += 1;

}

\*s.top++ = e;

s.m ++;

s.data[s.m] = e;

}//入栈

int Pop(SqStack &s)

{

int e;

e = s.m;

if (s.top == s.base) exit(0);

e = \*--s.top;

s.m--;

return e;

}//出栈

int Emptys(SqStack &s)

{

if (s.m==-1) return 1;

else return 0;

}

void process(int k, int a[], int x, SqStack &s)

{

int i, j;

if (k < n)

{

Push(s, k + 1);

process(k + 1, a, x, s);

Pop(s);

}

if (!Emptys(s))

{

i = Pop(s);

a[x] = i;

x++;

process(k, a, x, s);

Push(s, i);

}

if (Emptys(s))

{

if (x == n)

{

for (j = 0; j < x; j++)

{

printf("%2d", a[j]);

}

printf("\n");

}

}

}//利用递归

void main()

{

SqStack s;

int a[MAX];

printf("输入要调度的车厢总数");

scanf\_s("%d", &n);

InitStack(s);

Push(s, 1);

printf("输出序列为:\n");

process(1, a, 0, s);

}

1. 用户使用说明
2. 运行，输入车厢节数；
3. 回车即可显示所有输出序列；
4. 退出；
5. 调试分析

程序能成功运行并得出正确结果。

6.测试结果

输入n=3，输出结果为：

321 231 213 132 123