**3\_6游艇出租问题**

#include <iostream>

#include <memory.h>

#include <stdio.h>

#define N 200

using namespace std;

int r[N][N]; //存储费用数据

int p[N][N]; //记录最小花费

int smallestFee(int start, int n); //求最小费用

int answer[N];

int main()

{

int n;

while(1 == scanf\_s("%d", &n))//读取数据

{

memset(answer, 0, sizeof(answer));

memset(p, 0, sizeof(p));

for (int i = 1; i <= n - 1; ++i)

{

for (int j = i + 1; j <= n; ++j)

scanf\_s("%d", &r[i][j]);

}

printf("最小费用: %d\n", smallestFee(1, n)); //输出最小费用

//输出方案

while (getchar() != '\n') //剔除后续字符，为下一次输入做准备

continue;

}

return 0;

}

/\*自底向上计算\*/

int smallestFee(int start, int n)

{

//这两层循环，由左至右，计算了在下次停靠归还游艇之前的最小费用

for (int i = 2; i <= n; ++i)

{

int x = 2 << 10;

int temp;

for (int j = 1; j < i; ++j)

{

//如果从1到j站的最小费用加上从j直接到i站的费用比之前的最优方案更优，则选择这种方案

if (r[j][i] + p[1][j] < x)

{

x = r[j][i] + p[1][j];

temp = j;

}

}

p[1][i] = x; //记录最少花费

answer[temp] = temp; //记录最优方案的归还站点

}

return p[start][n];

}

**4-1会场安排问题**

#include<iostream>

using namespace std;

#define M 50 //最大活动数

struct Active

{

int s;//开始时间

int e;//结束时间

int no;//预安排会场号

}a[M];

//两元素交换位置

void swap(Active &a,Active &b)

{

Active t;

t=a;

a=b;

b=t;

}

void main()

{

int k;

int i,j;

cout<<"输入待安排活动数:"<<endl;

cin>>k;

cout<<"输入待安排活动的开始时间和结束时间:"<<endl;

//输入活动时间

for(i=1;i<=k;i++)

{

cin>>a[i].s>>a[i].e;

a[i].no=0;

}

//活动时间排序

for(i=1;i<=k;i++)

{

for(j=i;j<=k;j++)

{

if(a[i].s>a[j].s)

swap(a[i],a[j]);

if(a[i].s==a[j].s)

{

if(a[i].e>a[j].e)

swap(a[i],a[j]);

}

}

}

int sum=1;//使用的会场数初始化

int n;

a[1].no=sum;

for(i=2;i<=k;i++)

{

for(n=1;n<i;n++)

{

if(a[n].no!=0&&a[n].e<=a[i].s)

{

a[i].no=a[n].no;

a[n].no=0;//已经安排过的活动就不再比较

break;

}

}

if(n==i)

{

sum+=1;

a[i].no=sum;

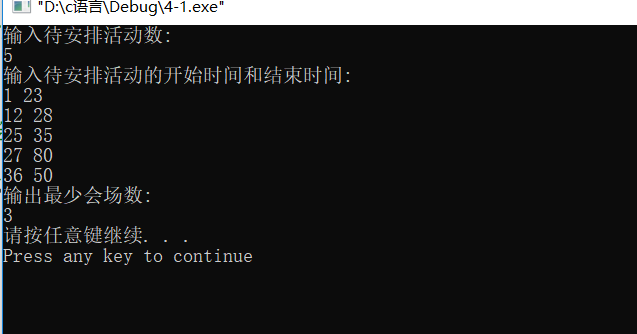
}

}

cout<<"输出最少会场数:\n"<<sum<<endl;

system("pause");

}



**5-2最小长度电路板排列问题**

#include<stdio.h>

#define n 8

#define m 5

int arr[n][m]={

{1,1,1,1,1},

{0,1,0,1,0},

{0,1,1,1,0},

{1,0,1,1,0},

{1,0,1,0,0},

{1,1,0,1,0},

{0,0,0,0,1},

{0,1,0,0,1}

};

int a[n],opt[n];

int mincount=99999;

int tempcount;

void traceback(int t)

{

int i,j;

int left,right;

int temp;

if(t==n)

{

tempcount=0;

for(i=0;i<m;i++)

{

for(j=0;j<n;j++) //最左边的

{

if(arr[a[j]][i]==1)

{

left=j;

break;

}

}

for(j=n-1;j>=0;j--) //最右边的

{

if(arr[a[j]][i]==1)

{

right=j;

break;

}

}

if(tempcount<right-left) //最大长度

{

tempcount=right-left;

}

}

if(tempcount<mincount) //最大长度的最小值

{

mincount=tempcount;

for(i=0;i<n;i++)

opt[i]=a[i];

}

return;

}

for(i=t;i<n;i++)

{

temp=a[i];

a[i]=a[t];

a[t]=temp;

traceback(t+1);

temp=a[i];

a[i]=a[t];

a[t]=temp;

}

}

int main()

{

int i;

for(i=0;i<n;i++)

a[i]=i;

traceback(0);

printf("%d\n",mincount);

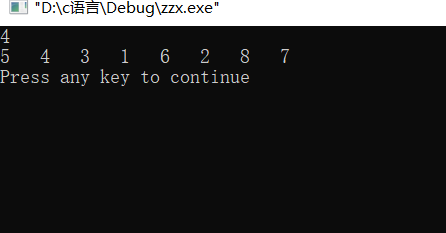
for(i=0;i<n;i++)

printf("%d ",opt[i]+1);

printf("\n");

return 0;

}



**5-8整数变换问题**

#include <iostream>

using namespace std;

int k = 1;

int c = 0;

char a[100] = {'\0'};

int SelectFun(const int n, const int m, int s) //选择函数

{

if(s == 0){

return 3 \* n;

}

else{

return n / 2;

}

}

bool DeptSearch(int Dept, const int n, const int m)//深搜

{

int num;

if(Dept > k) return false;

num = n;

for(int i = 0; i < 2; i++)

{

num = SelectFun(n, m, i);

if(num == m || DeptSearch(Dept + 1,num,m)){

if(i == 0){

a[c] = 'f';

}

else{

a[c] = 'g';

}

c ++;

return true;

}

}

return false;

}

int main()

{

int m, n, Dept = 1;

cin >> m >> n;

k = 1;

while( !DeptSearch(1, m, n) )

{

k ++;

}

cout << k << endl;

int i = 0;

for(i = 0; i < c; i ++){

cout << a[i];

}

return 0;

}

