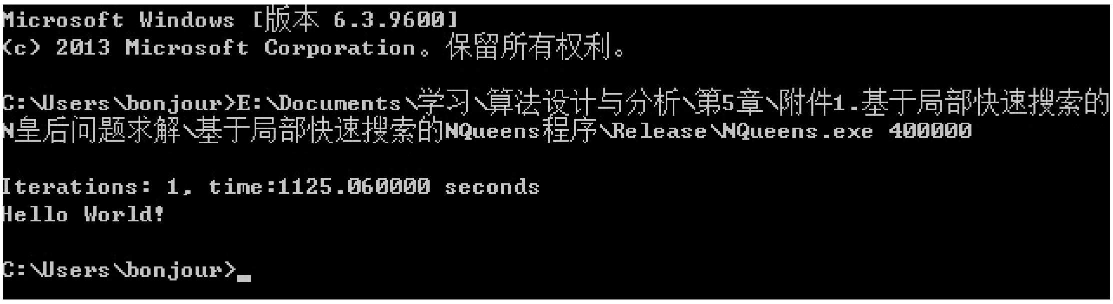
算法设计与分析——第五章作业

----学院 班级：-------- 姓名：--------- 学号：--------

● 运行结果

一、 N 皇后问题



```
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation。保留所有权利。

C:\Users\bonjour>E:\Documents\学习\算法设计与分析\第5章\附件1.基于局部快速搜索的
N皇后问题求解\基于局部快速搜索的NQueens程序\Release\NQueens.exe 400000

Iterations: 1, time:1125.060000 seconds
Hello World!

C:\Users\bonjour>_
```

图 1 400000 皇后运行结果

n	2000	5000	10000	20000	50000	100000	200000	400000
m	1	1	1	1	1	1	1	1
t(n)	0.015	0.062	0.265	1.078	8.622	37.048	150.570	1125.060
lg t(n)	-1.824	-1.208	-0.577	0.032	0.935	1.569	2.178	3.051
2 *lg* n	6.602	7.398	8	8.602	9.398	10	10.602	11.204
3 *lg* n	9.903	11.097	12	12.903	14.097	15	15.903	16.806

表 1 N 皇后问题的运行结果

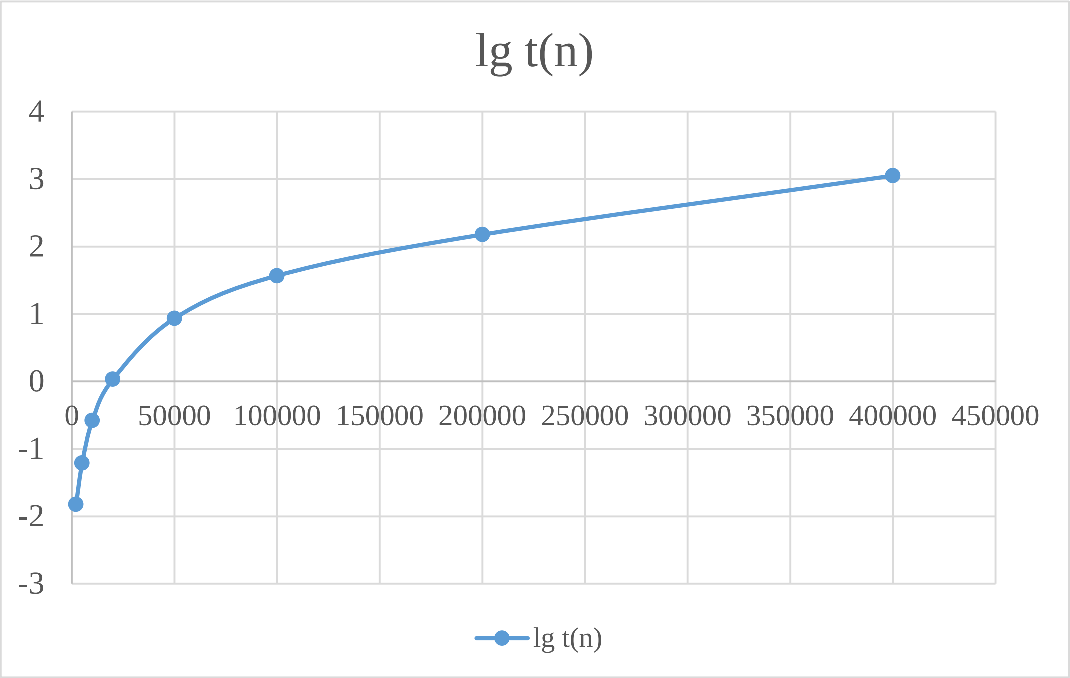


图 2　N 皇后问题中时间 Log t(n)和数量 n 之间的关系

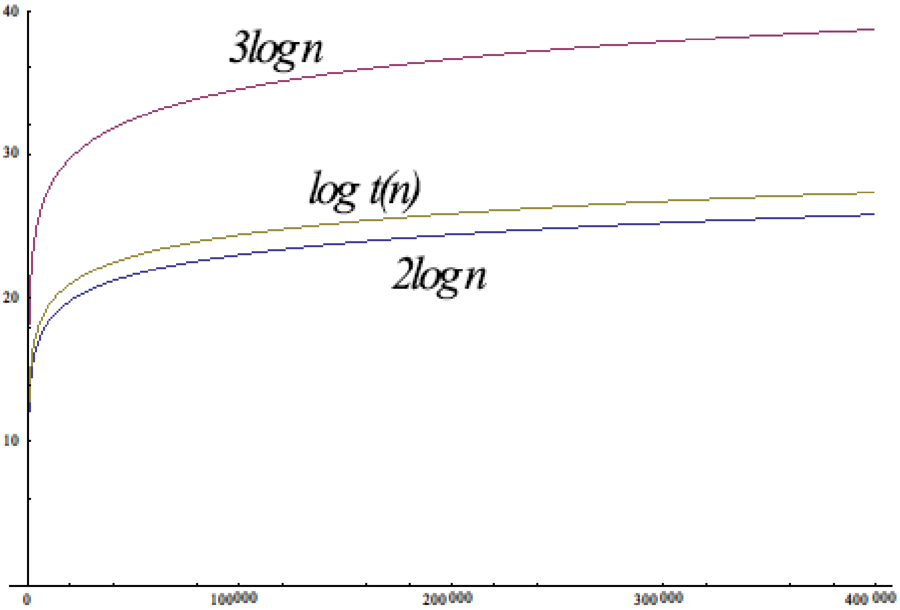


图 3　N 皇后问题中时间 log t(n)和数量 2log n、3log n 之间的关系

经过 Mathematica 拟合以后得到 log t(n)的曲线为：

$$20.70080791550514 + 2.1190267389563893\mathrm{Log}[x]$$

由图 3 可以看出 2<k<3，经过拟合后得到结果$k \approx 2.12$，即算法的时间复杂度为 O($n^{2.12}$)规模。

二、 图的 M 着色问题

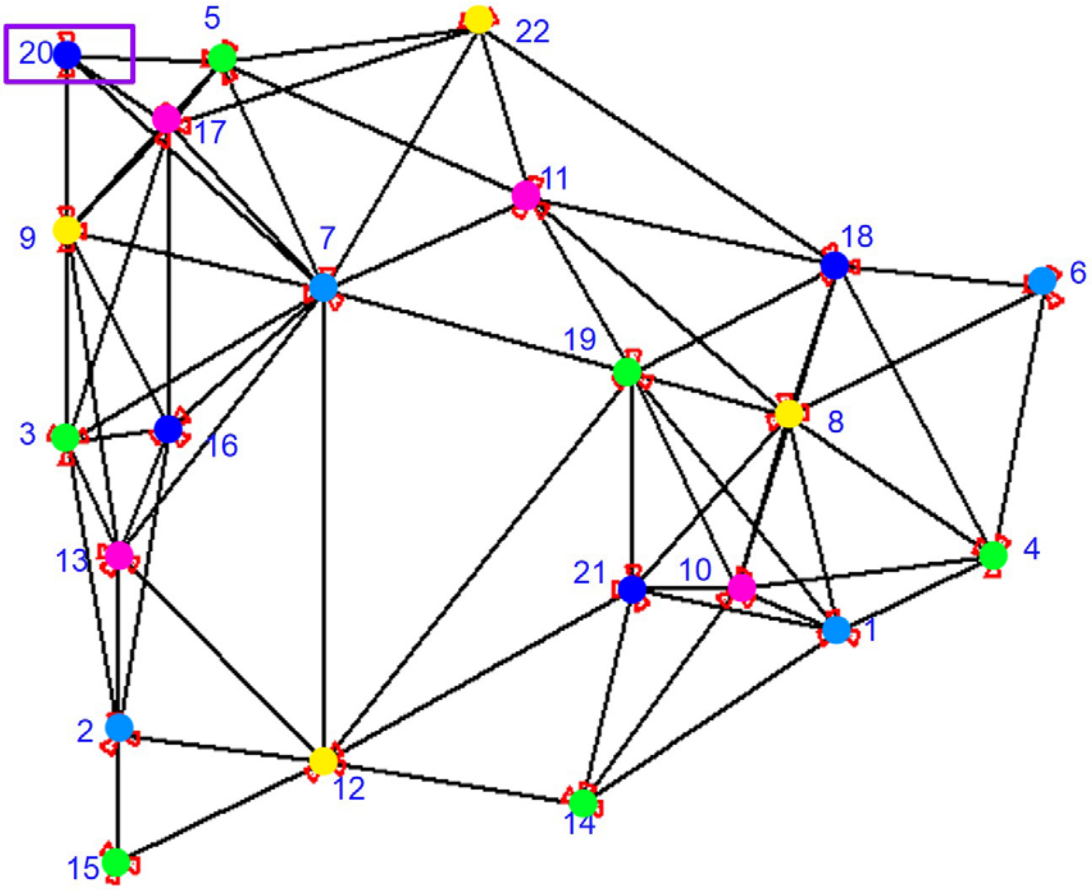
针对 22 个顶点构成的图：

最少需要 5 个颜色。

着色方案总数为 1426560 种。

运行总时间为 11029 毫秒。

遍历的节点总数 47028611 个。



程序输出的结果为：

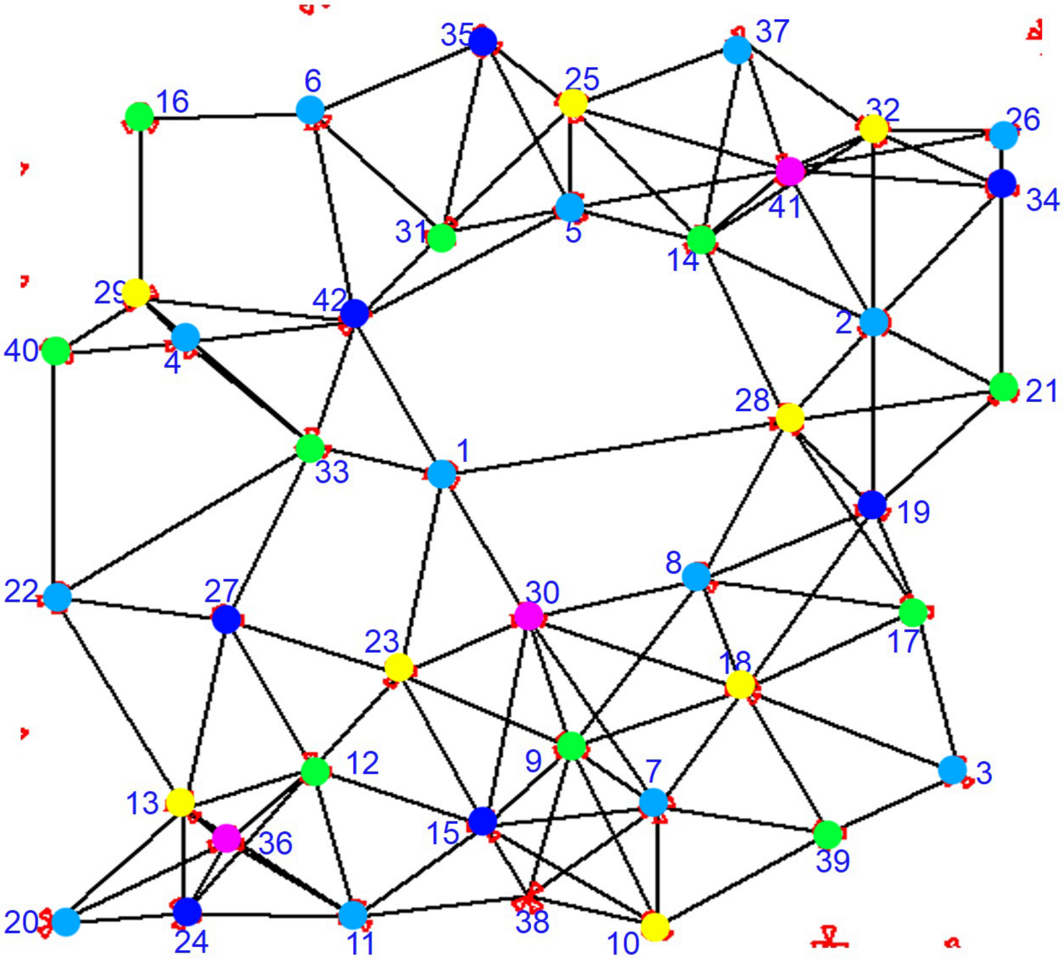
```
1:No Ans.
The time is 0
2:No Ans.
The time is 0
3:No Ans.
The time is 0
4:No Ans.
```

```
The time is 31
The minimum number of color is 5
The sum is 1426560.And the time is 10998
The nodes travelled:47028611
The color of node 1 is 1               The color of node 12 is 3
The color of node 2 is 1               The color of node 13 is 4
The color of node 3 is 2               The color of node 14 is 2
The color of node 4 is 2               The color of node 15 is 2
The color of node 5 is 2               The color of node 16 is 5
The color of node 6 is 1               The color of node 17 is 4
The color of node 7 is 1               The color of node 18 is 5
The color of node 8 is 3               The color of node 19 is 2
The color of node 9 is 3               The color of node 20 is 5
The color of node 10 is 4              The color of node 21 is 5
The color of node 11 is 4              The color of node 22 is 3
```

针对 42 个顶点构成的图：

最少需要 5 个颜色。

运行总时间为 1.94992x10^6 毫秒。



程序输出的结果：

```
1:No Ans.
The time is 0
2:No Ans.
The time is 0
3:No Ans.
The time is 37
4:No Ans.
The time is 1.94992e+006
The color of node 1is 1             The color of node 11is 1
The color of node 2is 1             The color of node 12is 2
The color of node 3is 1             The color of node 13is 3
The color of node 4is 1             The color of node 14is 2
The color of node 5is 1             The color of node 15is 4
The color of node 6is 1             The color of node 16is 2
The color of node 7is 1             The color of node 17is 2
The color of node 8is 1             The color of node 18is 3
The color of node 9is 2             The color of node 19is 4
The color of node 10is 3            The color of node 20is 1
The color of node 21is 2
The color of node 22is 1
The color of node 23is 3
The color of node 24is 4
The color of node 25is 3
The color of node 26is 1
The color of node 27is 4
The color of node 28is 3
The color of node 29is 3
The color of node 30is 5
The color of node 31is 2
The color of node 32is 3
The color of node 33is 2
The color of node 34is 4
The color of node 35is 4
The color of node 36is 5
The color of node 37is 1
The color of node 38is 5
The color of node 39is 2
The color of node 40is 2
The color of node 41is 5
The color of node 42is 4
```

三、 旅行商问题

针对 22 个顶点构成的图：

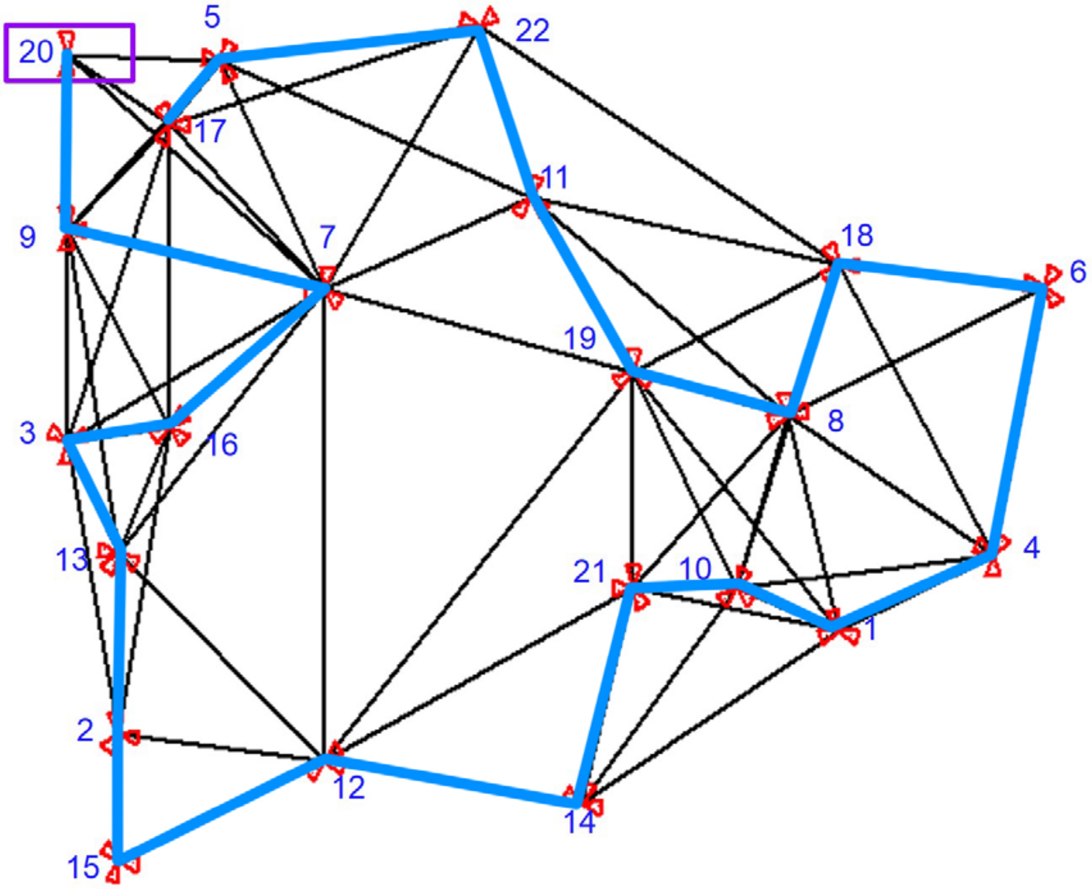
最短路径为：

20->9->7->16->3->13->2->15->12->14->21->10->1->4->6->18->8->19->11->22->5->17

其总长度为：7690.8

程序运行总时间为：26932 毫秒

总遍历节点数：405818247



程序输出的结果为：

```
The shortest length is 7690.8
The time cost is 26932
The node travelled: 405818247
The road is:
20->9->7->16->3->13->2->15->12->14->21->10->1->4->6->18->8->19->11->22->5
->17
```

● 源程序代码

二、 图的 M 着色问题

```
#include <iostream>
#include <cstdbool>
#include <cstdio>
#include <cmath>
#include <ctime>
#include "libxl.h" //用于读取excel文件

using namespace libxl; //用于读取excel文件
using namespace std;

#define NUM1 22 //第一个文件是22个基站的邻接矩阵
#define NUM2 42

int colorNum; //颜色数量
int arc[50][50]; //节点间是否连接
int ans[50]; //可行解
long int sum; //可行解总数量
long long int nodeNum; //遍历节点数
bool isFirst; //是否是第一个可行解

void backtrack(int t, int num);
bool ok(int k);

//数据和文件处理
int readData(Book* book, int arc[][50], wchar_t
loadFileName[], int sheetNum, int n);

int main(void){
    //从excel文件中读取数据
    Book* book = xlCreateBook();
    if (!book){
        cout << "Error when init book." << endl;
        return -1;
    }

//首先是22个顶点构成的图
    //读取数据
    wchar_t loadFileName1[] = L"附件1-1.基站图的邻接矩阵-v1.xls";
    if (readData(book, arc, loadFileName1, 1, NUM1) <= 0)
        return -2;

    clock_t start, finish;
    double duration;

    for (colorNum = 1; colorNum <= NUM1; colorNum++){
        //初始化
        isFirst = true;
        sum = 0;
        nodeNum = 0;
        start = clock();
        //开始遍历
        backtrack(1, NUM1);
        finish = clock();
        duration = (double)(finish - start);
        //输出结果
        if (sum != 0){
            cout << "The minimum number of color is " << colorNum
<< endl;
            cout << "The sum is " << sum << ".And the time is "
<< duration << endl;
            cout << "The nodes travelled:" << nodeNum << endl <<
endl;
            break;
        }
        else{
            cout << colorNum << ':' << "No Ans." << endl;
            cout << "The time is " << duration << endl;
        }
    }

//尝试遍历42个顶点
    Book* book2 = xlCreateBook();
    if (!book2){
        cout << "Error when init book." << endl;
        return -1;
    }
    if (readData(book2, arc, loadFileName1, 0, NUM2) <= 0)
        return -2;

    for (colorNum = 1; colorNum <= NUM2; colorNum++){
        sum = 0;
        nodeNum = 0;
        start = clock();
        backtrack(1, NUM2);
        finish = clock();
        duration = (double)(finish - start);
        if (sum != 0){
            cout << "The minimum number of color is " << colorNum
<< endl;
            cout << "The sum is " << sum << ".And the time is "
<< duration << endl;
            cout << "The nodes travelled:" << nodeNum << endl;
            break;
        }
        else{
            cout << colorNum << ':' << "No Ans." << endl;
            cout << "The time is " << duration << endl;
        }
    }

    system("PAUSE");
    return 0;
}

void backtrack(int t, int num){ //对所有节点进行回溯染色
    nodeNum++;
    if (t > num){ //如果t>num，说明所有的点都已经正确染色，此时输出结果
        sum++;
        if (isFirst){
            for (int j = 1; j <= NUM1; j++){
                cout << "The color of node " << j << "is " <<
ans[j] << endl;
            }
            isFirst = false;
        }
    }
    else{ //此时还有点未被正确着色
        for (int i = 1; i <= colorNum; i++){
            ans[t] = i; //将第t个点染成第i色
            if (ok(t)) //如果不与前t个点冲突，说明可行，继续染下一个点
                backtrack(t + 1, num);
            ans[t] = 0;//将这个点的颜色置空以回溯
        }
    }
}

bool ok(int k){ //判断第k个节点的颜色是否与前k-1个节点冲突
    for (int j = 1; j < k; j++)
```

```cpp
        if (arc[k][j] && (ans[j] == ans[k]))
            return false;
    return true;
}

//数据与文件处理
int readData(Book* book, int arc[][50], wchar_t
loadFileName[], int sheetNum, int n){ //将数据从excel文件中读出
    if (book->load(loadFileName)){ //读取book
        cout << "已读取文件。" << endl;
    }
    else{
        cout << "读取文件时错误。" << endl;
        return -2;
    }
    Sheet* sheet = book->getSheet(sheetNum);//读取excel文件中的
sheet2
    if (sheet){ //将sheet中的数据复制到结构数组中
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++){
                arc[i + 1][j + 1] = (int)sheet->readNum(i + 2, j +
2);
                if (arc[i + 1][j + 1] == -1)
                    arc[i + 1][j + 1] = 0;
                else
                    arc[i + 1][j + 1] = 1;
            }
    }
    book->release();
    return 1;
}
```

三、 旅行商问题

```
#include <iostream>
#include <cstdbool>
#include <cstdio>
#include <cmath>
#include <ctime>
#include "libxl.h" //用于读取excel文件

using namespace libxl; //用于读取excel文件
using namespace std;

#define NUM1 22 //第一个文件是22个基站的邻接矩阵

double arc[50][50];
int ans[50];
int bestAns[50];
double value;
double bestValue;
int nodeNum;

void tsp(int num);
void backtrack(int i, int num);


//数据和文件处理
int readData(Book* book, double arc[][50], wchar_t
loadFileName[], int sheetNum, int n);

int main(void){
    //从excel文件中读取数据
    Book* book = xlCreateBook();
    if (!book){
        cout << "Error when init book." << endl;
        return -1;
    }

    //读取数据
    wchar_t loadFileName1[] = L"附件1-1.基站图的邻接矩阵-v1.xls";
    if (readData(book, arc, loadFileName1, 1, NUM1) <= 0)
        return -2;

    //计时
    clock_t start, finish;
    double duration;

    start = clock();
    tsp(NUM1);
    finish = clock();
    duration = (double)(finish - start);

    //输出结果
    cout << "The shortest length is " << bestValue << endl;
    cout << "The time cost is " << duration << endl;
    cout << "The node travelled: " << nodeNum << endl;
    cout << "The road is: ";
    for (int i = 1; i <= NUM1; i++){
        cout << bestAns[i] << "->";
    }
    cout << endl;

    system("PAUSE");
    return 0;
}

void tsp(int num){ //初始化变量并开始回溯过程
    for (int i = 1; i <= num; i++)
        ans[i] = i;
    bestValue = 10000;
    value = 0;
    nodeNum = 0;
    backtrack(2, num);
}

void backtrack(int i, int num){ //回溯遍历各个节点
    nodeNum++;
    int temp;
    if (i == num){ //如果满足条件，则将这个结果记为最优结果
        if (arc[ans[num - 1]][ans[num]] < 10000 &&
            arc[ans[num]][1] < 10000 &&
            (bestValue == 10000 || value + arc[ans[num -
1]][ans[num]] + arc[ans[num]][1] < bestValue)){
            for (int j = 1; j <= num; j++)
                bestAns[j] = ans[j];
            bestValue = value + arc[ans[num - 1]][ans[num]] +
arc[ans[num]][1];
        }
    }
    else{ //如果不满足条件，交换i、j向下一个节点回溯
        for (int j = i; j <= num; j++)
            if (arc[ans[i - 1]][ans[j]] < 10000 && (bestValue ==
10000 || value + arc[ans[i - 1]][ans[j]] < bestValue)){
                temp = ans[i];
                ans[i] = ans[j];
                ans[j] = temp;

                value += arc[ans[i - 1]][ans[i]];
                backtrack(i + 1, num);
                value -= arc[ans[i - 1]][ans[i]];

                temp = ans[i];
                ans[i] = ans[j];
                ans[j] = temp;
            }
    }
}

//数据与文件处理
int readData(Book* book, double arc[][50], wchar_t
loadFileName[], int sheetNum, int n){ //将数据从excel文件中读出
    if (book->load(loadFileName)){ //读取book
        cout << "已读取文件。" << endl;
    }
    else{
        cout << "读取文件时错误。" << endl;
        return -2;
    }
    Sheet* sheet = book->getSheet(sheetNum);//读取excel文件中的
sheet2
    if (sheet){ //将sheet中的数据复制到结构数组中
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++){
                arc[i + 1][j + 1] = sheet->readNum(i + 2, j + 2);
                if (arc[i + 1][j + 1] == -1)
                    arc[i + 1][j + 1] = 10000;
            }
    }
    book->release();
    return 1;
}
```