

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

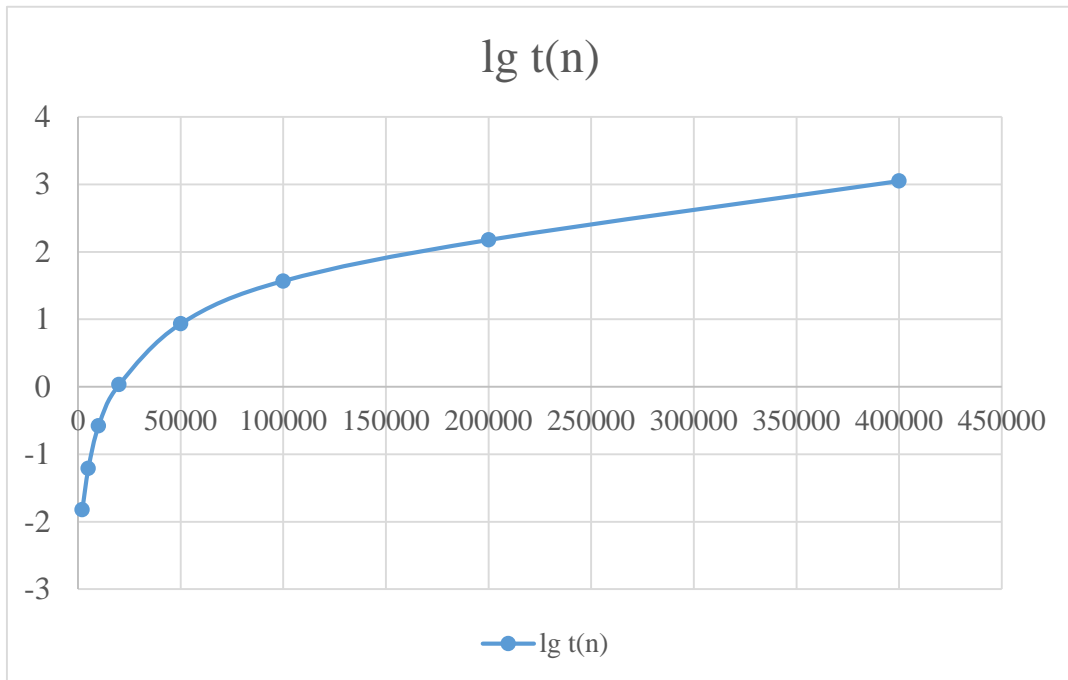


图 2 N 皇后问题中时间 $\text{Log } t(n)$ 和数量 n 之间的关系

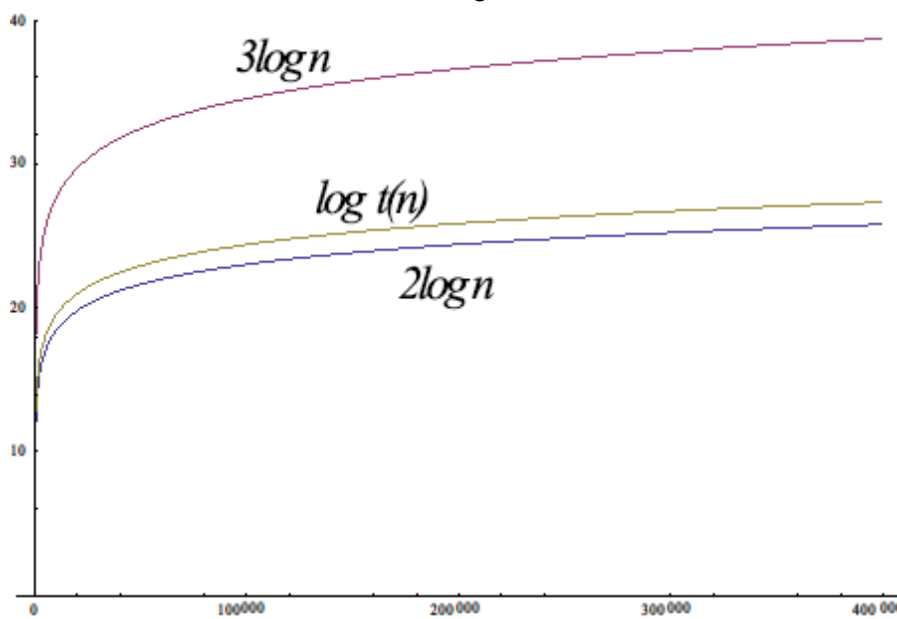


图 3 N 皇后问题中时间 $\log t(n)$ 和数量 $2\log n$ 、 $3\log n$ 之间的关系

经过 Mathematica 拟合以后得到 $\log t(n)$ 的曲线为：

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

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

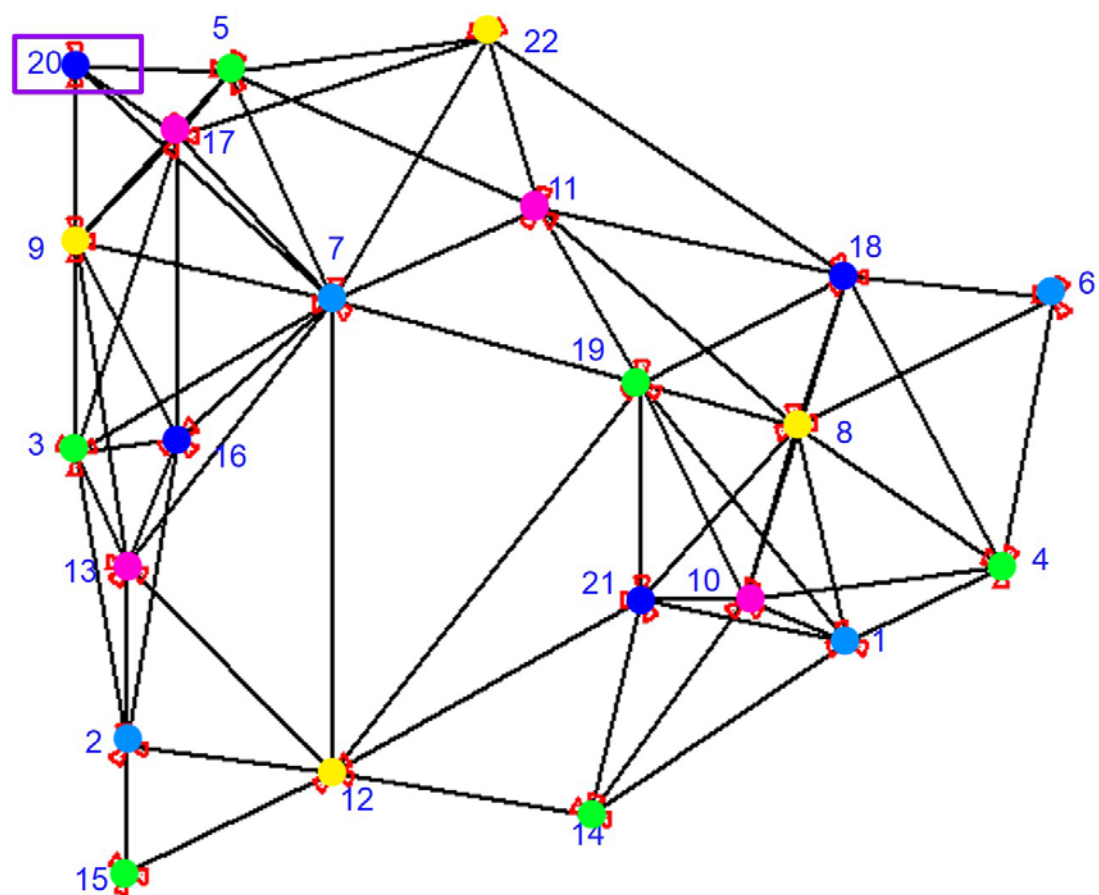
针对 22 个顶点构成的图：

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 2 is 1

The color of node 3 is 2

The color of node 4 is 2

The color of node 5 is 2

The color of node 6 is 1

The color of node 7 is 1

The color of node 8 is 3

The color of node 9 is 3

The color of node 10 is 4

The color of node 11 is 4

The color of node 12 is 3

The color of node 13 is 4

The color of node 14 is 2

The color of node 15 is 2

The color of node 16 is 5

The color of node 17 is 4

The color of node 18 is 5

The color of node 19 is 2

The color of node 20 is 5

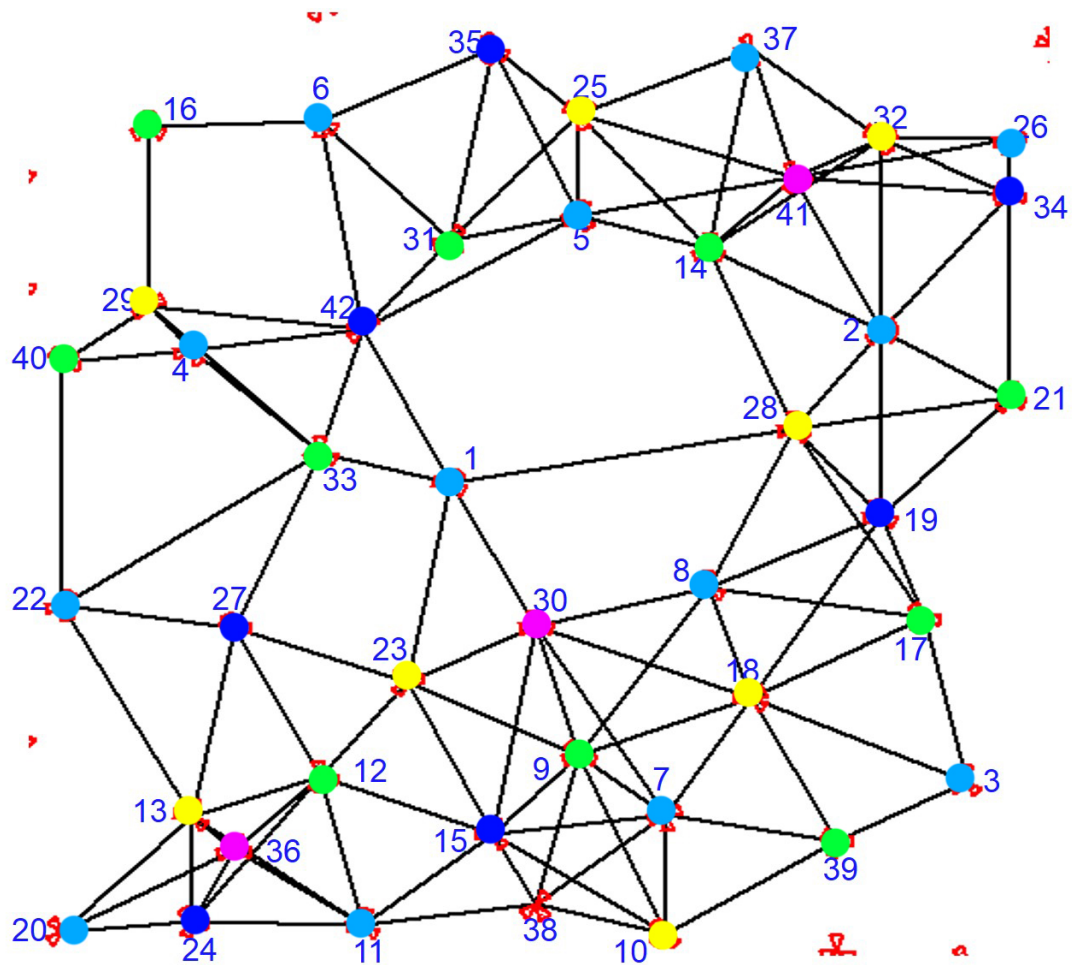
The color of node 21 is 5

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 1.94992×10^6 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 1 is 1

The color of node 2 is 1

The color of node 3 is 1

The color of node 4 is 1

The color of node 5 is 1

The color of node 6 is 1

The color of node 7 is 1

The color of node 8 is 1

The color of node 9 is 2

The color of node 10 is 3

The color of node 11 is 1

The color of node 12 is 2

The color of node 13 is 3

The color of node 14 is 2

The color of node 15 is 4

The color of node 16 is 2

The color of node 17 is 2

The color of node 18 is 3

The color of node 19 is 4

The color of node 20 is 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：

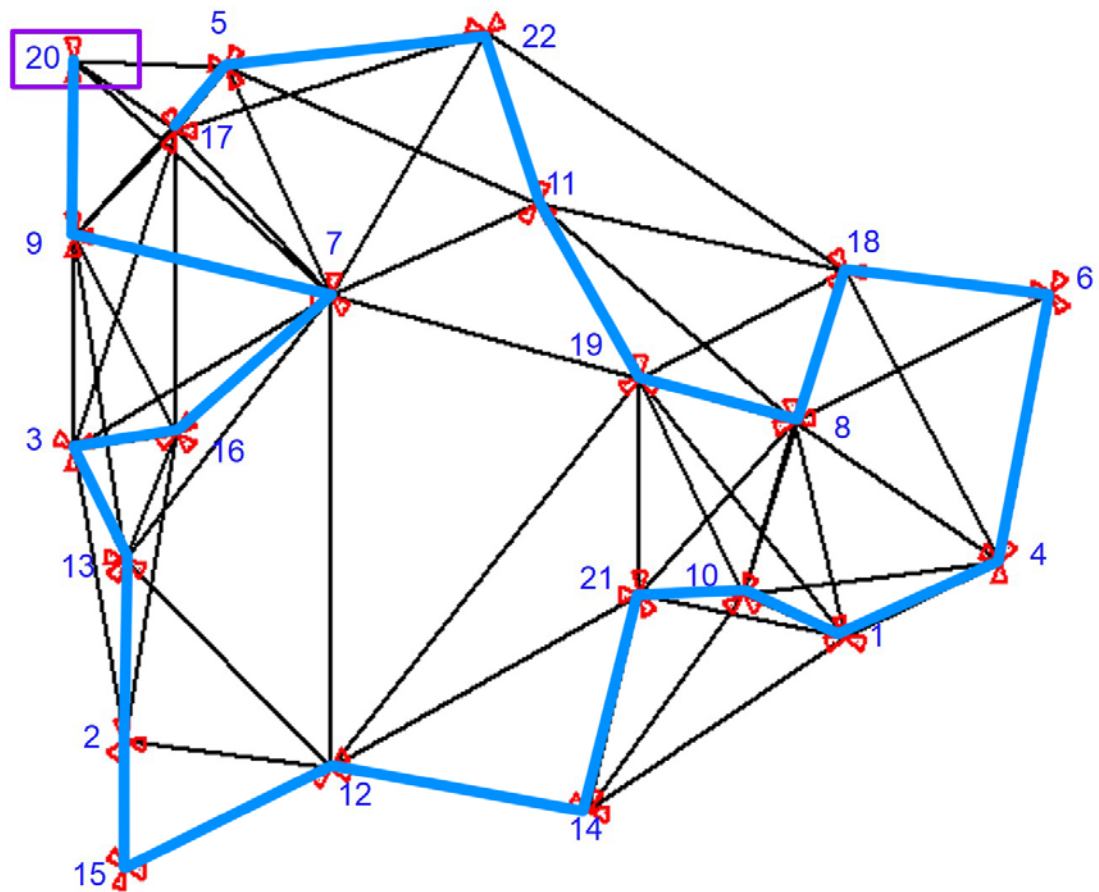
最短路径为：

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

```
1  #include <iostream>
2  #include <cstdbool>
3  #include <cstdio>
4  #include <cmath>
5  #include <ctime>
6  #include "libxl.h" //用于读取excel文件
7
8  using namespace libxl; //用于读取excel文件
9  using namespace std;
10
11 #define NUM1 22 //第一个文件是22个基站的邻接矩阵
12 #define NUM2 42
13
14 int colorNum; //颜色数量
15 int arc[50][50]; //节点间是否连接
16 int ans[50]; //可行解
17 long int sum; //可行解总数量
18 long long int nodeNum; //遍历节点数
19 bool isFirst; //是否是第一个可行解
20
21 void backtrack(int t, int num);
22 bool ok(int k);
23
24 //数据和文件处理
25 int readData(Book* book, int arc[][50], wchar_t
26 loadFileName[], int sheetNum, int n);
27
28 int main(void){
29     //从excel文件中读取数据
30     Book* book = xlCreateBook();
31     if (!book){
32         cout << "Error when init book." << endl;
33         return -1;
34     }
35
36     //首先是22个顶点构成的图
37     //读取数据
38     wchar_t loadFileName1[] = L"附件1-1.基站图的邻接矩阵-v1.xls";
39     if (readData(book, arc, loadFileName1, 1, NUM1) <= 0)
40         return -2;
```



```

41
42     clock_t start, finish;
43     double duration;
44
45     for (colorNum = 1; colorNum <= NUM1; colorNum++){
46         //初始化
47         isFirst = true;
48         sum = 0;
49         nodeNum = 0;
50         start = clock();
51         //开始遍历
52         backtrack(1, NUM1);
53         finish = clock();
54         duration = (double)(finish - start);
55         //输出结果
56         if (sum != 0){
57             cout << "The minimum number of color is " << colorNum
58 << endl;
59             cout << "The sum is " << sum << ".And the time is "
60 << duration << endl;
61             cout << "The nodes travelled:" << nodeNum << endl <<
62 endl;
63             break;
64         }
65         else{
66             cout << colorNum << ':' << "No Ans." << endl;
67             cout << "The time is " << duration << endl;
68         }
69     }
70
71     //尝试遍历42个顶点
72     Book* book2 = xlCreateBook();
73     if (!book2){
74         cout << "Error when init book." << endl;
75         return -1;
76     }
77     if (readData(book2, arc, loadFileName1, 0, NUM2) <= 0)
78         return -2;
79
80     for (colorNum = 1; colorNum <= NUM2; colorNum++){
81         sum = 0;
82         nodeNum = 0;
83         start = clock();
84         backtrack(1, NUM2);

```

```

85         finish = clock();
86         duration = (double)(finish - start);
87         if (sum != 0){
88             cout << "The minimum number of color is " << colorNum
89 << endl;
90             cout << "The sum is " << sum << ".And the time is "
91 << duration << endl;
92             cout << "The nodes travelled:" << nodeNum << endl;
93             break;
94         }
95         else{
96             cout << colorNum << ':' << "No Ans." << endl;
97             cout << "The time is " << duration << endl;
98         }
99     }
100
101     system("PAUSE");
102     return 0;
103 }
104
105 void backtrack(int t, int num){ //对所有节点进行回溯染色
106     nodeNum++;
107     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
108         sum++;
109         if (isFirst){
110             for (int j = 1; j <= NUM1; j++){
111                 cout << "The color of node " << j << "is " <<
112 ans[j] << endl;
113             }
114             isFirst = false;
115         }
116     }
117     else{ //此时还有点未被正确着色
118         for (int i = 1; i <= colorNum; i++){
119             ans[t] = i; //将第t个点染成第i色
120             if (ok(t)) //如果不与前t个点冲突, 说明可行, 继续染下一个点
121                 backtrack(t + 1, num);
122             ans[t] = 0; //将这个点的颜色置空以回溯
123         }
124     }
125 }
126
127 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
128     for (int j = 1; j < k; j++)

```

```

129         if (arc[k][j] && (ans[j] == ans[k]))
130             return false;
131     return true;
132 }
133
134 //数据与文件处理
135 int readData(Book* book, int arc[][50], wchar_t
136 loadFileName[], int sheetNum, int n){ //将数据从excel文件中读出
137     if (book->load(loadFileName)){ //读取book
138         cout << "已读取文件。" << endl;
139     }
140     else{
141         cout << "读取文件时错误。" << endl;
142         return -2;
143     }
144     Sheet* sheet = book->getSheet(sheetNum); //读取excel文件中的
145     sheet2
146     if (sheet){ //将sheet中的数据复制到结构数组中
147         for (int i = 0; i < n; i++)
148             for (int j = 0; j < n; j++){
149                 arc[i + 1][j + 1] = (int)sheet->readNum(i + 2, j +
150 2);
151                 if (arc[i + 1][j + 1] == -1)
152                     arc[i + 1][j + 1] = 0;
153                 else
154                     arc[i + 1][j + 1] = 1;
155             }
156     }
157     book->release();
158     return 1;
159 }

```

三、旅行商问题

```
1  #include <iostream>
2  #include <cstdbool>
3  #include <cstdio>
4  #include <cmath>
5  #include <ctime>
6  #include "libxl.h" //用于读取excel文件
7
8  using namespace libxl; //用于读取excel文件
9  using namespace std;
10
11 #define NUM1 22 //第一个文件是22个基站的邻接矩阵
12
13 double arc[50][50];
14 int ans[50];
15 int bestAns[50];
16 double value;
17 double bestValue;
18 int nodeNum;
19
20 void tsp(int num);
21 void backtrack(int i, int num);
22
23
24 //数据和文件处理
25 int readData(Book* book, double arc[][50], wchar_t
26 loadFileName[], int sheetNum, int n);
27
28 int main(void){
29     //从excel文件中读取数据
30     Book* book = xlCreateBook();
31     if (!book){
32         cout << "Error when init book." << endl;
33         return -1;
34     }
35
36     //读取数据
37     wchar_t loadFileName1[] = L"附件1-1.基站图的邻接矩阵-v1.xls";
38     if (readData(book, arc, loadFileName1, 1, NUM1) <= 0)
39         return -2;
40
41     //计时
42     clock_t start, finish;
```

```

43     double duration;
44
45     start = clock();
46     tsp(NUM1);
47     finish = clock();
48     duration = (double)(finish - start);
49
50     //输出结果
51     cout << "The shortest length is " << bestValue << endl;
52     cout << "The time cost is " << duration << endl;
53     cout << "The node travelled: " << nodeNum << endl;
54     cout << "The road is: ";
55     for (int i = 1; i <= NUM1; i++){
56         cout << bestAns[i] << "->";
57     }
58     cout << endl;
59
60     system("PAUSE");
61     return 0;
62 }
63
64 void tsp(int num){ //初始化变量并开始回溯过程
65     for (int i = 1; i <= num; i++)
66         ans[i] = i;
67     bestValue = 10000;
68     value = 0;
69     nodeNum = 0;
70     backtrack(2, num);
71 }
72
73 void backtrack(int i, int num){ //回溯遍历各个节点
74     nodeNum++;
75     int temp;
76     if (i == num){ //如果满足条件，则将这个结果记为最优结果
77         if (arc[ans[num - 1]][ans[num]] < 10000 &&
78             arc[ans[num]][1] < 10000 &&
79             (bestValue == 10000 || value + arc[ans[num -
80 1]][ans[num]] + arc[ans[num]][1] < bestValue)){
81             for (int j = 1; j <= num; j++)
82                 bestAns[j] = ans[j];
83             bestValue = value + arc[ans[num - 1]][ans[num]] +
84             arc[ans[num]][1];
85         }
86     }

```

```

87     else{ //如果不满足条件, 交换i、j向下一个节点回溯
88         for (int j = i; j <= num; j++)
89             if (arc[ans[i - 1]][ans[j]] < 10000 && (bestValue ==
90 10000 || value + arc[ans[i - 1]][ans[j]] < bestValue)){
91                 temp = ans[i];
92                 ans[i] = ans[j];
93                 ans[j] = temp;
94
95                 value += arc[ans[i - 1]][ans[i]];
96                 backtrack(i + 1, num);
97                 value -= arc[ans[i - 1]][ans[i]];
98
99                 temp = ans[i];
100                ans[i] = ans[j];
101                ans[j] = temp;
102            }
103    }
104 }
105
106 //数据与文件处理
107 int readData(Book* book, double arc[][50], wchar_t
108 loadFileName[], int sheetNum, int n){ //将数据从excel文件中读出
109     if (book->load(loadFileName)){ //读取book
110         cout << "已读取文件。" << endl;
111     }
112     else{
113         cout << "读取文件时错误。" << endl;
114         return -2;
115     }
116     Sheet* sheet = book->getSheet(sheetNum); //读取excel文件中的
117     sheet2
118     if (sheet){ //将sheet中的数据复制到结构数组中
119         for (int i = 0; i < n; i++)
120             for (int j = 0; j < n; j++){
121                 arc[i + 1][j + 1] = sheet->readNum(i + 2, j + 2);
122                 if (arc[i + 1][j + 1] == -1)
123                     arc[i + 1][j + 1] = 10000;
124             }
125     }
126     book->release();
127     return 1;
128 }
129

```