

# 浙江工业大学

## 算法分析与设计实验报告

(2021 级)



实验题目

实验 1

学生姓名

温家伟

学生学号

202103151422

专业班级

大数据分析 2101

所在学院

理学院

提交日期

2023-3-8

### 第5章 回溯法

#### 实验结果及相应代码

# 1. 旅行售货员

## 1.1 PTA提交代码截图

提交结果

2

题目	用户	提交时间
7-1	猫圣钧	2023/05/14 10:05:52
编译器	内存	用时
C++ (g++)	456 / 65536 KB	4 / 400 ms
状态	分数	评测时间
答案正确	10 / 10	2023/05/14 10:05:53

评测详情

测试点	提示	内存(KB)	用时(ms)	结果	得分
0		452	4	答案正确	5 / 5
1		456	4	答案正确	5 / 5

提交代码

复制内容

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 const int N=105;
4 int g[N][N];
5 int x[N];
```

## 1.2 PTA提交代码结果截图

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  const int N=105;
4  int g[N][N];
5  int x[N];
6  int n;
7  int ans=0x3f3f3f3f;
8  int now=0;
9  void TSP(int u)
10 {
11     if(u>n)
12     {
13         if(now+g[x[n]][x[1]]<ans)
14         {
15             ans=now+g[x[n]][x[1]];
16         }
17     }
18     else
19     {
20         for(int i=u;i<=n;i++)
21         {
22             if(g[x[u-1]][x[i]]!=0x3f3f3f3f&&now+g[x[u-1]]
23             {
24                 swap(x[u],x[i]);
25                 now+=g[x[u-1]][x[u]];
26                 TSP(u+1);
27                 now-=g[x[u-1]][x[u]];
28                 swap(x[u],x[i]);
29             }
30         }
31     }
32 }
33 int main()
34 {
35     cin>>n;
36     memset(g,0x3f,sizeof(g));
37     for(int i=1;i<=n;i++)
38     {
39         for(int j=1;j<=n;j++)

```

```
40         {  
41             cin>>g[i][j];  
42         }  
43     }  
44     for(int i=1;i<=n;i++) x[i]=i;  
45  
46     TSP(2);  
47     cout<<ans;  
48     return 0;  
49 }  
50
```

### 1.3 算法分析

使用回溯法可以枚举所有情况（全排列），从而求出最优解。

- 将起始点作为当前点。
- 从当前点开始，前往没有遍历过的点。
- 在第一次回到起始点时，判断是否遍历了所有点。
- 如果是，则更新最优解，然后返回上一个节点（回溯到上一个分支）。
- 如果不是，则继续前往未访问过的节点。
- 直到所有情况都枚举完毕。

## 2 装载问题

### 2.1 PTA提交代码截图

C++ (g++)

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  const int MAX_N = 100;
7
8  int c1, c2;
9  int n;
10 vector<int> w(MAX_N);
11 vector<int> flag(MAX_N);
12 int rest_capacity = 0;
13 int current_weight = 0;
14 int optimal_weight = 0;
15 vector<int> choose(MAX_N);
16
17 void backtrack(int i) {
18     ... if (i > n) {
19         ... if (current_weight > optimal_weight) {
20             ... optimal_weight = current_weight;
21             ... choose.assign(flag.begin(), flag.end());
22             ... }
23         ... return;
24     ... }
25
26     ... rest_capacity -= w[i];
27
28     ... if (current_weight + w[i] <= c1) {
29         ... current_weight += w[i];
30         ... flag[i] = 1;
31         ... backtrack(i + 1);
32         ... current_weight -= w[i];
33         ... flag[i] = 0;
34     ... }
35
36     ... if (current_weight + rest_capacity > optimal_weight) {
37         ... backtrack(i + 1);
38     ... }
39
```

```

40     ...rest_capacity += w[i];
41 }
42
43 int main() {
44     ...int sum = 0;
45     ...cin >> n;
46
47     ...for (int i = 1; i <= n; i++) {
48         ...cin >> w[i];
49         ...rest_capacity += w[i];
50         ...sum += w[i];
51     ...}
52
53     ...cin >> c1 >> c2;
54
55     ...backtrack(1);
56
57     ...int loop_count = 0;
58     ...if (sum - optimal_weight <= c2) {
59         ...for (int i = 1; i <= n; i++) {
60             ...if (choose[i] == 1) {
61                 ...if (!loop_count) {
62                     ...cout << i;
63                     ...loop_count++;
64                 } else {
65                     ...cout << " " << i;
66                 }
67             ...}
68         ...}
69         ...loop_count = 0;
70         ...cout << endl;
71
72         ...for (int i = 1; i <= n; i++) {
73             ...if (choose[i] == 0) {
74                 ...if (!loop_count) {
75                     ...cout << i;
76                     ...loop_count++;
77                 } else {
78                     ...cout << " " << i;
79                 }
80             ...}

```

```
81     .....}
82     ....}.else.{
83     .....cout<< "No solution"<< endl;
84     ....}
85
86     ....return 0;
87 }
88
```

2.2 PTA提交代码结果截图

提交结果

题目	用户	提交时间
7-2	猫圣钧	2023/05/14 10:18:03
编译器	内存	用时
C++ (g++)	456 / 65536 KB	5 / 400 ms
状态 ?	分数	评测时间
答案正确	30 / 30	2023/05/14 10:18:03

评测详情					
测试点	提示	内存(KB)	用时(ms)	结果	得分
1		332	5	答案正确	10 / 10
2		328	5	答案正确	10 / 10
3		456	4	答案正确	10 / 10

提交代码

复制内容

```
1 #include <iostream>
2 #include <vector>
3
```

2.3 算法分析

- 将所有货物从大到小排序。
- 初始化两个卡车的当前重量为0，当前位置为1。
- 从第一个货物开始，依次尝试将其放到两个卡车中的任意一个上，如果可以，则更新对应卡车的当前重量，并进入下一个货物的选择；如果不行，则回溯到上一个状态。
- 当所有货物都尝试过后，比较两辆卡车的重量差，更新最优解。
- 如果当前的状态是合法状态（即两辆卡车的重量都没有超过总载重量的一半），则继续扩展状态；否则，回溯到上一个状态。
- 当所有状态都扩展完毕后，回溯到最初状态，输出最优解。