

重 庆 大 学

学 生 实 验 报 告

实验课程名称 算法设计与分析

开课实验室 DS1501

学 院 大数据与软件学院 年级 2021 专业班 软件工程 X 班

学 生 姓 名 XXX 学 号 2021XXXX

开 课 时 间 2022 至 2023 学年第 二 学期

总 成 绩	
教师签名	XXX

《算法设计与分析》实验报告

开课实验室：DS1501

2023 年 5 月 16 日

学院	大数据与软件学院	年级、专业、班	2021 级软件 工程 X 班	姓名	XXX	成绩	
课程 名称	算法设计与分析		实验项目 名 称	蛮力法实验和回溯法实 验		指导教师	XXX
教师 评 语	<div>教师签名：_____</div> <div>年 月 日</div>						
<div><div>一、实验目的</div><div><ul style="list-style-type: none">● 掌握蛮力法和回溯法的设计思想，包括蛮力法的基本应用、回溯法的算法框架等。● 熟练掌握“钱币兑换问题求解算法”的实现，熟练掌握“填字游戏问题求解算法”的实现，提供输入案例检测算法的正确性。● 主要任务：实现教材配套实验指导书“第 2 章 2.4.2 小节 求解钱币兑换问题”和“第 2 章 2.5.2 小节 求解填字游戏问题”。</div><div><div>二、使用仪器、材料</div><div>PC 微机 Lenovo Legion R9000P2021H; Windows11 操作系统; Clion2023 编译环境;</div><div><div>三、实验步骤</div><div><div>2.4.2 实验 2 求解钱币兑换问题</div><div><p>某个国家仅有 1 分、2 分和 5 分硬币,将钱 $n(n \geq 5)$ 兑换成硬币有很多种兑法。编写一个实验程序计算出 10 分钱有多少种兑法,并列出每种兑换方式。</p><p>解: (蛮力法)</p><p>设钱 n 兑换成 1 分、2 分、5 分的个数分别为 x、y、z,得到一个等式 $n=1 \times x+2 \times y+5 \times z$。在一次兑换中最多有 $z=n/5$(取整)个 5 分钱币,在余下的钱的兑换中最多有 $y=(n-5z)/2$(取整)个 2 分钱币,再把余下的钱兑换成 x 个 1 分钱币。</p></div><div><div>2.5.2 实验 2 求解填字游戏问题</div><div><p>在 3×3 个方格的方阵中要填入数字 1~10 的某 9 个数字, 每个方格填一个整数,使所有相邻两个方格内的两个整数之和为素数。编写一个实验程序,求出所有满足这个要求的数字填法。</p></div></div></div></div></div></div>							

解：（回溯法）

首先是 3×3 个方格的布局,采用一个一维数组 $a[9]$ 存放一种填字方案,与 3×3 个方格 b 的对应关系如图 2. 24 所示,即 $b[0][0]$ 存放在 $a[0]$ 中、 $b[0][1]$ 存放在 $a[1]$ 中... $b[2][2]$ 存放在 $a[8]$ 中。在这种存储结构中,如何确定 a 的 i 位置的相邻方格呢?采用向前判断的方式:

- a 的 0 位置没有前向的相邻方格;
- a 的 1 位置的前向的相邻方格是 $a[0]$;
- a 的 2 位置的前向的相邻方格是 $a[1]$;
- a 的 3 位置的前向的相邻方格是 $a[0]$;
- a 的 4 位置的前向的相邻方格是 $a[1]$ 和 $a[3]$;
- a 的 5 位置的前向的相邻方格是 $a[2]$ 和 $a[4]$;
- a 的 6 位置的前向的相邻方格是 $a[3]$;
- a 的 7 位置的前向的相邻方格是 $a[4]$ 和 $a[6]$;
- a 的 8 位置的前向的相邻方格是 $a[5]$ 和 $a[7]$ 。

	0	1	2
0	0	1	2
1	3	4	5
2	6	7	8

图 2. 24 3×3 个方格 b
采用一维数组
 a 表示

为此,采用一个二维数组 $Checkmatrix$ 表示相邻方格如下(a 的 i 位置的相邻方格对应第 i 行,以-1 表示结尾);

另外,用 $used[1..10]$ 表示对应数字是否使用过, $used[i]=true$ 表示数字 i 没有使用,后面可以使用它填字, $used[i]=false$ 表示数字 i 已经填入,不能再使用它。

四、实验过程原始记录(数据、图表、计算等)

2.4.2 实验 2 求解钱币兑换问题

某个国家仅有 1 分、2 分和 5 分硬币,将钱 $n(n \geq 5)$ 兑换成硬币有很多种兑法。编写一个实验程序计算出 10 分钱有多少种兑法,并列出每种兑换方式。

源代码:

```
#include <iostream>

void resolve(int total){
    int n1,n2,n5,count; // 定义三种硬币的数量以及兑换方案的数量
    count = 0; // 初始兑换方案数量为 0
    for(n5=0;n5<=total/5;n5++){ // 遍历 5 分硬币数量,从 0 开始,一直到总数可以被 5 整除的那个数字
        for(n2=0;n2<=(total-5*n5)/2;n2++){ // 遍历 2 分硬币数量,从 0 开始,一直到剩下的总数可以被 2 整除的那个数字
            if(5*n5+2*n2<=total){ // 如果当前的硬币总价值不超过总数
                n1=total-5*n5-2*n2; // 计算出 1 分硬币的数量
                count++; // 兑换方案数量加 1
                std::cout<<"兑法"<<count<<"为: ";<<std::endl;
                if(n5>0){
                    std::cout<<"5分硬币"<<n5<<"个 ";
                }
                if(n2>0){
                    std::cout<<"2分硬币"<<n2<<"个 ";
                }
                if(n1>0){
                    std::cout<<"1分硬币"<<n1<<"个";
                }
                std::cout<<std::endl; // 输出一个换行符
            }
        }
    }

    std::cout<<"共有"<<count<<"种兑法,具体如上所述。"<<std::endl; // 输出总的兑换方案数以及具体的每种兑换方案
}

int main(){
    int total;
    total=10; // 设置要找零的总数为 10 分
    std::cout<<"2.4.2问题求解结果为: ";<<std::endl;
    resolve(total); // 调用 resolve 函数进行找零计算
    return 0; // 程序正常结束并返回 0
}
```

运行结果：

2.4.2问题求解结果为：

兑法1为：

1分硬币10个

兑法2为：

2分硬币1个 1分硬币8个

兑法3为：

2分硬币2个 1分硬币6个

兑法4为：

2分硬币3个 1分硬币4个

兑法5为：

2分硬币4个 1分硬币2个

兑法6为：

2分硬币5个

兑法7为：

5分硬币1个 1分硬币5个

兑法8为：

5分硬币1个 2分硬币1个 1分硬币3个

兑法9为：

5分硬币1个 2分硬币2个 1分硬币1个

兑法10为：

5分硬币2个

共有10种兑法，具体如上所述。

进程已结束,退出代码0

2.5.2 实验2 求解填字游戏问题

在 3×3 个方格的方阵中要填入数字 1~10 的某 9 个数字, 每个方格填一个整数, 使所有相邻两个方格内的两个整数之和为素数。编写一个实验程序, 求出所有满足这个要求的数字填法。

源代码:

```
#include <iostream>
#include <cmath>
#include <iomanip>

#define N 10 // 定义常量 N

bool used[N+1]; // 标记数组, 用于标记数字是否已被使用
int a[9]; // 存储当前解的数组
int count = 0; // 记录解的总数
int check_matrix[][3] = { [0]: { [0]: -1, [1]: { [0]: 0, [1]: -1, [2]: { [0]: 1, [1]: -1, [3]: { [0]: 0, [1]: -1, [4]: { [0]: 1, [1]: 3, [2]: -1, [5]: { [0]: 2, [1]: 4, [2]: -1, [6]: { [0]: 3, [1]: -1, [7]: { [0]: 4, [1]: 6, [2]: -1, [8]: { [0]: 5, [1]: 7, [2]: -1}};

// 输出解
void out_solution(int a[]){
    int i, j;
    std::cout<<"第"<<count<<"个解为: ";<<std::endl;
    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            std::cout<<std::setw(3)<<a[3*i+j];
        }
        std::cout<<std::endl;
    }
}

// 判断一个数是否为素数
bool if_prime(int m){
    for(int i=2; i<=sqrt((X) m); i++){
        if(m%i==0){
            return false;
        }
    }
    return true;
}
```

```
// 检查当前解是否符合条件
bool check(int p){
    int i, j;
    if(p<0){
        return false;
    }
    for(i=0; (j=check_matrix[p][i])>=0; i++){
        if(!if_prime((m: a[p]+a[j]))){
            return false;
        }
    }
    return true;
}
```

```
// 在未使用的数字中选择第一个数字
int select_num(int start){
    for(int j=start;j<=N;j++){
        if(used[j]){
            return j;
        }
    }
    return 0;
}

// 在当前位置扩展数字
int add(int p){
    a[++p]=select_num( start: 1);
    used[a[p]]= false;
    return p;
}

// 回溯操作, 寻找下一个可行的解
int change(int p){
    int j;
    while(p>=0 && (j= select_num( start: a[p]+1))==0){
        used[a[p--]]=true;
    }
    if(p<0){
        return -1;
    }
    used[a[p]]= true;
    a[p]=j;
    used[j]= false;
    return p;
}
```

```
// 解决问题
void resolve(){
    bool flag=true;
    int p=0;
    a[p]=1; // 初始解为 1
    used[a[p]]= false;
```

```

do{
    if(flag){ // 如果当前解符合条件，尝试扩展
        if(p==8){ // 找到一个解后输出，并回溯寻找下一个解
            out_solution(a);
            p= change(p);
        }
        else{
            p= add(p);
        }
    }
    else{ // 如果当前解不符合条件，回溯
        p=change(p);
    }
    flag = check(p); // 检查当前解是否符合条件
} while (p>=0);
}

int main(){
    for(int i=1;i<=N;i++){ // 初始化标记数组，所有数字均未使用
        used[i]=true;
    }
    std::cout<<"对该问题求解如下： "<<std::endl;
    resolve(); // 解决问题
    std::cout<<"所以解的总数为： "<<std::endl;
    std::cout<<"count= "<<count<<std::endl; // 输出解的总数
    return 0;
}

```

运行结果：

```

对该问题求解如下：
第1个解为：
1  2  5
4  3  8
7  10 9
第2个解为：
1  2  5
4  9  8
7  10 3
第3个解为：
1  2  5
10 3  8
7  4  9

```


第4个解为：

1 2 5

10 9 8

7 4 3

第5个解为：

1 4 7

2 3 10

5 8 9

第6个解为：

1 4 7

2 9 10

5 8 3

第7个解为：

1 10 7

2 3 4

5 8 9

第8个解为：

1 10 7

2 9 4

5 8 3

第9个解为：

2 1 6

3 4 7

8 9 10

第10个解为：

2 1 6

3 10 7

8 9 4

第11个解为：

2 1 6

9 4 7

8 3 10

第12个解为：

2 1 6

9 10 7

8 3 4

第13个解为：

2 3 8

1 4 9

6 7 10

第14个解为：

2 3 8

1 10 9

6 7 4

第15个解为：

2 9 8

1 4 3

6 7 10

第16个解为：

2 9 8

1 10 3

6 7 4

第17个解为：

3 2 5

4 1 6

9 10 7

第18个解为：

3 2 5

10 1 6

9 4 7

第19个解为：

3 4 7

8 9 10

5 2 1

第20个解为：

3 4 7

10 1 6

9 2 5

第21个解为：

3 4 9

2 1 10

5 6 7

第22个解为：

3 4 9

10 1 2

7 6 5

第23个解为：

3 8 5

4 9 2

7 10 1

第24个解为：

3 8 5

10 9 2

7 4 1

第25个解为：

3 10 7

4 1 6

9 2 5

第26个解为：

3 10 7

8 9 4

5 2 1

第27个解为：

3 10 9

2 1 4

5 6 7

第28个解为：

3 10 9

4 1 2

7 6 5

第29个解为：

4 1 6

3 2 5

10 9 8

第30个解为：

4 1 6

9 2 5

10 3 8

第31个解为：

4 3 8

7 10 9

6 1 2

第32个解为：

4 3 8

9 2 5

10 1 6

第33个解为：

4 3 10

1 2 9

6 5 8

第 34 个解为：

4 3 10

9 2 1

8 5 6

第 35 个解为：

4 7 6

3 10 1

8 9 2

第 36 个解为：

4 7 6

9 10 1

8 3 2

第 37 个解为：

4 9 8

3 2 5

10 1 6

第 38 个解为：

4 9 8

7 10 3

6 1 2

第 39 个解为：

4 9 10

1 2 3

6 5 8

第 40 个解为：

4 9 10

3 2 1

8 5 6

第 41 个解为：

5 2 1

8 3 4

9 10 7

第 42 个解为：

5 2 1

8 3 10

9 4 7

第 43 个解为：

5 2 1

8 9 4

3 10 7

第 44 个解为：

5 2 1

8 9 10

3 4 7

第 45 个解为：

5 2 3

6 1 4

7 10 9

第 46 个解为：

5 2 3

6 1 10

7 4 9

第 47 个解为：

5 2 9

6 1 4

7 10 3

第 48 个解为：

5 2 9

6 1 10

7 4 3

第 49 个解为：

5 6 7

2 1 4

3 10 9

第 50 个解为：

5 6 7

2 1 4

9 10 3

第 51 个解为：

5 6 7

2 1 10

3 4 9

第 52 个解为：

5 6 7

2 1 10

9 4 3

第 53 个解为：

5 8 3

2 9 4

1 10 7

第 54 个解为：

5 8 3

2 9 10

1 4 7

第 55 个解为：

5 8 9

2 3 4

1 10 7

第 56 个解为：

5 8 9

2 3 10

1 4 7

第 57 个解为：

6 1 2

7 4 3

10 9 8

第 58 个解为：

6 1 2

7 4 9

10 3 8

第 59 个解为：

6 1 2

7 10 3

4 9 8

第 60 个解为：

6 1 2

7 10 9

4 3 8

第 61 个解为：

6 1 4

5 2 3

8 9 10

第 62 个解为：

6 1 4

5 2 9

8 3 10

第 63 个解为：

6 1 10

5 2 3

8 9 4

第 64 个解为：

6 1 10

5 2 9

8 3 4

第 65 个解为：

6 5 8

1 2 3

4 9 10

第 66 个解为：

6 5 8

1 2 3

10 9 4

第 67 个解为：

6 5 8

1 2 9

4 3 10

第 68 个解为：

6 5 8

1 2 9

10 3 4

第 69 个解为：

6 7 4

1 10 3

2 9 8

第 70 个解为：

6 7 4

1 10 9

2 3 8

第 71 个解为：

6 7 10

1 4 3

2 9 8

第 72 个解为：

6 7 10

1 4 9

2 3 8

第 73 个解为：

7 4 1

10 3 2

9 8 5

第 74 个解为：

7 4 1

10 9 2

3 8 5

第 75 个解为：

7 4 3

6 1 10

5 2 9

第 76 个解为：

7 4 3

10 9 8

1 2 5

第 77 个解为：

7 4 9

6 1 10

5 2 3

第 78 个解为：

7 4 9

10 3 8

1 2 5

第 79 个解为：

7 6 5

4 1 2

3 10 9

第 80 个解为：

7 6 5

4 1 2

9 10 3

第 81 个解为：

7 6 5

10 1 2

3 4 9

第 82 个解为：

7 6 5

10 1 2

9 4 3

第 83 个解为：

7 10 1

4 3 2

9 8 5

第 84 个解为：

7 10 1

4 9 2

3 8 5

第 85 个解为：

7 10 3

4 9 8

1 2 5

第 86 个解为：

7 10 3

6 1 4

5 2 9

第 87 个解为：

7 10 9

4 3 8

1 2 5

第 88 个解为：

7 10 9

6 1 4

5 2 3

第 89 个解为：

8 3 2

9 4 1

10 7 6

第 90 个解为：

8 3 2

9 10 1

4 7 6

第 91 个解为：

8 3 4

5 2 9

6 1 10

第 92 个解为：

8 3 4

9 10 7

2 1 6

第 93 个解为：

8 3 10

5 2 9

6 1 4

第 94 个解为：

8 3 10

9 4 7

2 1 6

第 95 个解为：

8 5 6

3 2 1

4 9 10

第 96 个解为：

8 5 6

3 2 1

10 9 4

第 97 个解为：

8 5 6

9 2 1

4 3 10

第 98 个解为：

8 5 6

9 2 1

10 3 4

第 99 个解为：

8 9 2

3 4 1

10 7 6

第 100 个解为：

8 9 2

3 10 1

4 7 6

第 101 个解为：

8 9 4

3 10 7

2 1 6

第 102 个解为：

8 9 4

5 2 3

6 1 10

第 103 个解为：

8 9 10

3 4 7

2 1 6

第 104 个解为：

8 9 10

5 2 3

6 1 4

第 105 个解为：

9 2 5

4 1 6

3 10 7

第 106 个解为：

9 2 5

10 1 6

3 4 7

第 107 个解为：

9 4 3

2 1 10

5 6 7

第 108 个解为：

9 4 3

10 1 2

7 6 5

第 109 个解为：

9 4 7

8 3 10

5 2 1

第 110 个解为：

9 4 7

10 1 6

3 2 5

第 111 个解为：

9 8 5

4 3 2

7 10 1

第 112 个解为：

9 8 5

10 3 2

7 4 1

第 113 个解为：

9 10 3

2 1 4

5 6 7

第 114 个解为：

9 10 3

4 1 2

7 6 5

第 115 个解为：

9 10 7

4 1 6

3 2 5

第 116 个解为：

9 10 7

8 3 4

5 2 1

第 117 个解为：

10 1 6

3 2 5

4 9 8

第 118 个解为：

10 1 6

9 2 5

4 3 8

第 119 个解为：

10 3 4

1 2 9

6 5 8

第 120 个解为：

10 3 4

9 2 1

8 5 6

第 121 个解为：

10 3 8

7 4 9

6 1 2

第 122 个解为：

10 3 8

9 2 5

4 1 6

第 123 个解为：

10 7 6

3 4 1

8 9 2

第 124 个解为：

10 7 6

9 4 1

8 3 2

第 125 个解为：

10 9 4

1 2 3

6 5 8

第 126 个解为：

10 9 4

3 2 1

8 5 6

第 127 个解为：

10 9 8

3 2 5

4 1 6

第 128 个解为：

10 9 8

7 4 3

6 1 2

所以解的总数为：

count= 128

五、实验结果及分析

结果都已对应显示在原始数据记录中，结果都与预期的分析符合。