

# 第8章 算法设计初体验

## 第8.2课 蛮力法

---

《信息学奥赛一本通·编程启蒙 C++版》

蛮力法（brute force method，也称为穷举法或枚举法）是一种简单直接地解决问题的方法，常常直接基于问题的描述，所以，蛮力法也是最容易应用的方法。

### （1）简单型枚举

简单枚举就是枚举一些例如整数、子串的简单类型。一般把所有答案的可能性放在 for 里，前面的章节已经讲过，所以这里不再重复叙述。

### （2）排列枚举

排列枚举就是数学上的排列组合或全排列，即将一个集合里的所有（ $n$  个）元素重新排列顺序的得到其它数列，判断生成的数列是否满足题目的要求。共有  $n!$  种。可以使用 STL 中提供全排列函数：

```
<1> prev_permutation(iterator start, iterator end, bool cmp) ;
```

```
<2> next_permutation(iterator start, iterator end, bool cmp);
```

### （3）子集枚举

子集枚举的方法是与而二进制码紧密挂钩。我们发现每个子集其实都可以和一串二进制对应一个整数。

## 【例 82.1】不吉利日期

### 【题目描述】

在国外，每月的 13 号和每周的星期 5 都是不吉利的。特别是当 13 号那天恰好是星期 5 时，更不吉利。已知某年的一月一日是星期  $w$ ，并且这一年一定不是闰年，求出这一年所有 13 号那天是星期 5 的月份，按从小到大的顺序输出月份数字。（ $w= 1..7$ ）

### 【输入格式】

输入有一行，即一月一日星期几( $w$ )。

### 【输出格式】

输出有一到多行，每行一个月份，表示该月的 13 日是星期五。

### 【样例输入】

7

### 【样例输出】

1

10

## 【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int w,d;
4. int main(){
5.     cin>>w;
6.     for(int i=1;i<=12;i++){
7.         if(i==1 || i==3 || i==5 || i==7 || i==8 || i==10 || i==12) d=31;
8.         else if(i==2) d=28;
9.         else d=30;
10.        for(int j=1;j<=d;j++){
11.            if(w%7==5&&j==13) cout<<i<<endl;
12.            w++;
13.        }
14.    }
15.}
```

## 【例 82.2】 座次问题

### 【题目描述】

小 A 的学校，老师好不容易解决了市赛的报名问题，现在老师又犯愁了。现在有 N 位同学参加比赛，但是老师想给他们排座位，但是排列方式太多了。老师非常想弄明白最后的排座次的结果是什么样子的，到底有多少种结果。

请设计一个程序帮助老师。最后输出各种情况的人名即可，一行一种情况，每种情况的名字按照报名即输入顺序排序。

### 【输入格式】

第一行 输入 N;

第二行 到 第 N+1 行 共输入 N 个人名。

由于小 A 学校承办能力实在有限，所以其中 N 小于等于 10 人。

### 【输出格式】

输出各种情况的人名即可，一行一种情况，每种情况的名字按照报名即输入顺序排序

### 【样例输入】

```
3
xiaowang
xiaoA
xiaoli
```

### 【样例输出】

```
xiaowang xiaoA xiaoli
xiaowang xiaoli xiaoA
xiaoA xiaowang xiaoli
xiaoA xiaoli xiaowang
xiaoli xiaowang xiaoA
xiaoli xiaoA xiaowang
```

## 【代码实现】

```
1. #include <bits/stdc++.h>
2. using namespace std;
3. int main(){
4.     vector<string>M1;
5.     vector<int>M2;
6.     int N;
7.     string name;
8.     cin >> N;
9.     for(int i = 0; i<N;i++){
10.        cin >> name;
11.        M1.push_back(name);
12.        M2.push_back(i);}
13.    do{
14.        for(vector<int> ::iterator it = M2.begin() ; it!= M2.end( ) ; it++){
15.            cout <<M1[(*it)]<<" ";
16.        }
17.        cout << endl;
18.    }while(next_permutation(M2.begin(),M2.end()));
19.    return 0;
20.}
```

## 【例 82.3】 公平抽签

### 【题目描述】

小明的学校，市赛的参赛名额非常有限，只有  $m$  个名额，但是共有  $n$  个人报名。

作为老师非常苦恼，他不知道该让谁去，他在寻求一个绝对公平的方式。

于是他准备让大家抽签决定，即  $m$  个签是去，剩下的是不去。

小明非常想弄明白最后的抽签结果会有多少种不同到情况，请你设计一个程序帮帮小明！

### 【输入格式】

输入第一行包含两个字符  $n, m$ ，其含义如题所述。

接下来第二行到第  $n+1$  行每行包含一个字符串  $S$ ，表示个人名。 $1 \leq m \leq n \leq 15$ 。

### 【输出格式】

输出共若干行，每行包含  $m$  个字符串，表示该结果被选中到人名。

同一个方案中按读入的人名顺序输出，不同方案的输出顺序任意。

### 【样例输入】

```
3 2
xiaowang
xiaoA
xiaoli
```

### 【样例输出】

```
xiaowang xiaoA
xiaowang xiaoli
xiaoA xiaoli
```

## 【代码实现】

```
1. #include <bits/stdc++.h>
2. using namespace std;
3. int n,m;
4. string a[20];
5. int main(){
6.     cin>>n>>m;
7.     for(int i=0;i<n;i++) cin>>a[i];
8.     int U=1<<n;
9.     for(int S=0;S<U;S++)
10.         if(__builtin_popcount(S)==m){
11.             for(int i=0;i<n;i++)
12.                 if(S & (1<<i)) cout<<a[i]<<' ';
13.             cout<<endl;
14.         }
15.     return 0;
16. }
```

## 练 82.1 涂国旗

### 【题目描述】

某国法律规定，只要一个由  $N \times M$  个小方块组成的旗帜符合如下规则，就是合法的国旗。

从最上方若干行（至少一行）的格子全部是白色的；

接下来若干行（至少一行）的格子全部是蓝色的；

剩下的行（至少一行）全部是红色的；

现有一个棋盘状的布，分成了  $N$  行  $M$  列的格子，每个格子是白色蓝色红色之一，小  $a$  希望把这个布改成该国国旗，方法是在一些格子上涂颜料，盖住之前的颜色。

小明很懒，希望涂最少的格子，使这块布成为一个合法的国旗。

### 【输入格式】

第一行是两个整数  $N, M$ 。对于 100% 的数据， $N, M \leq 50$

接下来  $N$  行是一个矩阵，矩阵的每一个小方块是 W（白），B（蓝），R（红）中的一个。

### 【输出格式】

一个整数，表示至少需要涂多少块

### 【样例输入】

4 5

WRWRW

BWRWB

WRWRW

RWBWR

### 【样例输出】

11

## 【代码实现】

```
1. #include <bits/stdc++.h>
2. using namespace std;
3. int n,m,ans,mi=2500; //mi 初始化成一个很大的数
4. char c[51][51];
5. int main()
6. {
7.     int i,j,k,g;
8.     cin>>n>>m;
9.     for(i=1;i<=n;i++)
10.         for(j=1;j<=m;j++) cin>>c[i][j];
11.     for(i=1;i<=n-2;i++) //由于白色下面还有蓝色和红色, 所以i (白
        与蓝的边界) 枚举到 (n-2)
12.         for(j=i+1;j<=n-1;j++) //j (蓝与红的边界) 至少要比i 大1,
            同理枚举到 (n-1), 这样可以减少枚举次数
13.         {
14.             ans=0; //初始化
15.             //枚举三个区域
16.             for(k=1;k<=i;k++)
17.                 for(g=1;g<=m;g++) if(c[k][g]!='W') ans++;
18.             for(k=i+1;k<=j;k++)
19.                 for(g=1;g<=m;g++) if(c[k][g]!='B') ans++;
20.             for(k=j+1;k<=n;k++)
21.                 for(g=1;g<=m;g++) if(c[k][g]!='R') ans++;
22.             mi=min(ans,mi); //更新答案
23.         }
24.     cout<<mi<<endl;
25.     return 0;
26. }
```

## 练 82.2 三连击

### 【题目描述】

将  $1, 2, \dots, 9$  共  $9$  个数分成三组，分别组成三个三位数，且使这三个三位数的比例是  $A:B:C$ ，试求出所有满足条件的三个三位数，若无解，输出 **No!!!**。

### 【输入格式】

三个数， $A, B, C$ 。保证  $A < B < C$

### 【输出格式】

若干行，每行  $3$  个数字。按照每行第一个数字升序排列

### 【样例输入】

1 2 3

### 【样例输出】

192 384 576

219 438 657

273 546 819

327 654 981

## 【代码实现】

```
1. #include <bits/stdc++.h>
2. using namespace std;
3. int a[10]={0,1,2,3,4,5,6,7,8,9}; //打表
4. long long A,B,C,x,y,z;
5. int main(){
6.     scanf("%d%d%d",&A,&B,&C);
7.     int cnt=0;
8.     do{
9.         x=100*a[1]+10*a[2]+a[3]; //第一个数
10.        y=100*a[4]+10*a[5]+a[6]; //第二个数
11.        z=100*a[7]+10*a[8]+a[9]; //第三个数
12.        if(x*B==y*A && y*C==z*B){ //判断
13.            cnt++;
14.            printf("%d %d %d\n",x,y,z); //输出
15.        }
16.    }
17.    while(next_permutation(a+1,a+10));
18.    if(cnt==0){
19.        printf("No!!!");
20.    }
21.    return 0;
22.}
```

## 练 82.3 选数

### 【题目描述】

已知  $n$  个整数  $x_1, x_2, \dots, x_n$ , 以及 1 个整数  $k$  ( $k < n$ )。从  $n$  个整数中任选  $k$  个整数相加, 可分别得到一系列的和。例如当  $n=4, k=3$ , 4 个整数分别为 3, 7, 12, 19 时, 可得全部的组合与它们的和为:

$$3+7+12=22$$

$$3+7+19=29$$

$$7+12+19=38$$

$$3+12+19=34$$

现在, 要求你计算出和为素数共有多少种。

例如上例, 只有一种的和为素数:  $3+7+19=29$ 。

### 【输入格式】

第一行为  $n$  和  $k$  ( $1 \leq n \leq 20, k < n$ )

第二行为  $n$  个数

$x_1 x_2 \dots x_n$  ( $1 \leq x_i \leq 5000000$ ), 各数之间用一个空格隔开。

### 【输出格式】

一个整数(满足条件的种数)

### 【样例输入】

4 3

3 7 12 19

### 【样例输出】

## 【代码实现】

```
1. #include <bits/stdc++.h>
2. using namespace std;
3. int n,k,ans;
4. int a[30];
5. bool check(int x){
6.     if(x<2) return 0;
7.     if(x==2) return 1;
8.     for(int i=2;i*i<=x;i++)
9.         if(x%i==0) return 0;
10.    return 1;
11.}
12.int main(){
13.    cin>>n>>k;
```

```
14.    for(int i=0;i<n;i++) cin>>a[i];
15.    int U=1<<n;
16.    for(int S=0;S<U;S++)
17.        if(__builtin_popcount(S)==k){
18.            int sum=0;
19.            for(int i=0;i<n;i++)
20.                if(S & (1<<i)) sum+=a[i];
21.            if(check(sum)) ans++;
22.        }
23.    cout<<ans;
24.    return 0;
25.}
```

**谢谢!**

—