

# ACM程序设计竞赛算法讲座

## 第四讲

沈 俊

852679238@QQ.COM

2020.10.25

## 暴力枚举（2）

---

### 学习目标

1. 熟练应用暴力枚举法解决一些实际问题。
2. 体验暴力枚举法的审题分析和细节测试。

# 例1、涂国旗

## 【问题描述】

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。（毛熊：阿嚏——）

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a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  $N, M$ 。

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。

# 例1、涂国旗

## 【输入样例】

4 5

WRWRW

BWRWB

WRWRW

RWBWR

## 【输出样例】

11

## 【样例说明】

WWWWW

BBBBB

RRRRR

RRRRR

```
#include<bits/stdc++.h>
using namespace std;

int n,m,ans=0x7fffffff,w[51],b[51],r[51];
string s;
inline int check(char c){
    int tot=0;
    for(int i=0;i<m;++i)
        if(s[i]!=c)++tot;
    return tot;
}
```

```
int main(int argc, char const *argv[])
{
    cin>>n>>m;
    for(int i=1;i<=n;++i){
        cin>>s;
        w[i]=w[i-1]+check('W');
        b[i]=b[i-1]+check('B');
        r[i]=r[i-1]+check('R');
    }
    for(int i=1;i<n-1;++i)
        for(int j=i+1;j<n;++j)
            ans=min(ans,w[i]+b[j]-b[i]+r[n]-r[j]);
    cout<<ans;
    return 0;
}
```

## 例2、火柴棒等式

### 【问题描述】

给你n根火柴棍，你可以拼出多少个形如“A+B=C”的等式？等式中的A、B、C是用火柴棍拼出的整数（若该数非零，则最高位不能是0）。用火柴棍拼数字0-9的拼法如图所示：



注意：

- 1、加号与等号各自需要两根火柴棍
- 2、如果 $A \neq B$ ，则 $A+B=C$ 与 $B+A=C$ 视为不同的等式 ( $A, B, C \geq 0$ )
- 3、n根火柴棍必须全部用上

### 【输入格式】

一个整数n。

### 【输出格式】

一个整数，表示能拼成的不同等式的数目。

### 【输入样例1】

14

### 【输出样例1】

2

### 【输入样例2】

18

### 【输出样例2】

9

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int a[2001]={6,2,5,5,4,5,6,3,7,6},n,ans=0,i,j;
    cin >> n;
    for(i=10;i<=2000;i++) {
        j = i;
        while(j>=1) {
            a[i]=a[i]+a[j%10];      j=j/10;
        }
    }
    for(i=0;i<=1000;i++) {
        for(j=0;j<=1000;j++)
            if((a[i]+a[j]+a[i+j]+4)==n) ans++;
    }
    cout << ans << endl;
    return 0;
}
```



## 例3、PERKET

### 【问题描述】

Perket 是一种流行的美食。为了做好 Perket，厨师必须谨慎选择食材，以在保持传统风味的同时尽可能获得最全面的味道。你有 $n$ 种可支配的配料。对于每一种配料，我们知道它们各自的酸度 $s$ 和苦度 $b$ 。当我们添加配料时，总的酸度为每一种配料的酸度总乘积；总的苦度为每一种配料的苦度的总和。

众所周知，美食应该做到口感适中，所以我们希望选取配料，以使得酸度和苦度的绝对差最小。

另外，我们必须添加至少一种配料，因为没有任何食物以水为配料的。

### 【输入格式】

第一行一个整数 $n$  ( $1 \leq n \leq 10$ )，表示可供选用的食材种类数。

接下来 $n$ 行，每行2个整数 $s_i$ 和 $b_i$ ，表示第 $i$ 种食材的酸度和苦度。

### 【输出格式】

一个整数，一行一个整数，表示可能的总酸度和总苦度的最小绝对差。

### 【输入样例1】

1

3 10

### 【输出样例1】

7

## 例2、PERKET

### 【输入样例2】

2  
3 8  
5 8

### 【输出样例2】

1

### 【输入样例3】

4  
1 7  
2 6  
3 8  
4 9

### 【输出样例2】

1

```
#include<bits/stdc++.h>
using namespace std;
```

```
const int M=11;
int s[M],b[M],n,ans=0x7fff;
void dfs(int i,int x,int y){
    if(i>n){
        if(x==1&& y==0)return;
        ans=min(abs(x-y),ans);
        return;
    }
    dfs(i+1,x*s[i],y+b[i]);
    dfs(i+1,x,y);
}
```

```
int main(){
    cin >> n;
    for(int i=1;i<=n;i++){
        cin >> s[i] >> b[i];
    }
    dfs(1,1,0);
    cout << ans << endl;
    return 0;
}
```

## 例4、回文质数

### 【问题描述】

因为151既是一个质数又是一个回文数（从左到右和从右到左是看一样的），所以151是回文质数。

写一个程序来找出范围  $[a, b]$  ( $5 \leq a < b \leq 100,000,000$ ) (一亿) 间的所有回文质数。

### 【输入格式】

第一行2个整数  $a$  和  $b$ 。

### 【输出格式】

输出一个回文质数的列表，一行一个。

### 【输入样例】

5 500

### 【输出样例1】

5  
7  
11  
101  
131  
151  
181  
191  
313  
353  
373  
383

```

#include<bits/stdc++.h>
using namespace std;
bool p[100000001];
void prime(int b) {
    memset(p, true, sizeof(p));
    p[1]=false;
    int n=sqrt(b);
    for (int i=2;i<=n;i++)
        if (p[i])
            for (int j=2;j<=b/i;j++)
                p[i*j]=false;
}
bool isPalindromes(int num) {
    int temp=num,ans=0;
    while (temp!=0) {
        ans=ans*10+temp%10;
        temp/=10;
    }
    if (ans==num)
        return true;
    else
        return false;
}

```

```

int main() {
    int a,b;
    cin>>a>>b;
    if (b>=100000000)
        b=99999999;
    prime(b);
    if (a%2==0) a++;
    for (int i=a;i<=b;i+=2) {
        if (p[i] && isPalindromes(i))
            cout<<i<<endl;
    }
    return 0;
}

```