项目完成报告

组合数学

——硬币中的组合数

作 者 姓 名： 张喆

学 号： 1754060

指 导 教 师： 冯巾松

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 分析 1](#_Toc532672542)

[1.1 题目说明 1](#_Toc532672543)

[1.2 背景分析 1](#_Toc532672544)

[1.3 解题思路 1](#_Toc532672545)

[2 算法设计 2](#_Toc532672546)

[2.1 穷举法 2](#_Toc532672547)

[2.1.1 穷举法核心思想 2](#_Toc532672548)

[2.1.2 穷举法核心代码 2](#_Toc532672549)

[2.1.3 穷举法运行截图 3](#_Toc532672550)

[2.1.4 穷举法优缺点分析 3](#_Toc532672551)

[2.2 动态规划法 4](#_Toc532672552)

[2.2.1 动态规划法核心思想 4](#_Toc532672553)

[2.2.2 动态规划法核心代码 4](#_Toc532672554)

[2.2.3 动态规划法运行截图 5](#_Toc532672555)

[2.2.4 动态规划法优缺点分析 6](#_Toc532672556)

[2.3 母函数法 7](#_Toc532672557)

[3.2.1 母函数法核心思想 7](#_Toc532672558)

[3.2.2 母函数法核心代码 7](#_Toc532672559)

[3.2.3 母函数法运行截图 9](#_Toc532672560)

[3.2.4 母函数法优缺点分析 11](#_Toc532672561)

[3 结果分析 12](#_Toc532672562)

[4 结论 12](#_Toc532672563)

# 1 分析

## 1.1 题目说明

比较三种货币（人民币、美元、欧元）中硬币构成各自货币里10元的组合数。对比说明每种硬币的表现力

人民币有六种面额的硬币，分别是1分、2分、5分、1角、5角、1元。

美元有六种面额的硬币，分别是1美分、5美分、10美分、25美分，50美分、1美元。

欧元有八种面额的硬币，分别是 1欧分、2欧分、5欧分、10欧分、20欧分、50欧分、1欧元和2欧元。

## 1.2 背景分析

硬币中的组合数项目是解决“不同国家发行的不同数量和面额的硬币构成各自货币中的10元的组合数”问题。

不同的硬币种类在组成同样的金额时的表现能力差别很大，这对国家和地区发行货币的种类和金额有很重要的意义。究竟什么样的组合可以让货币的表性能力更好，这是一个值得我们去研究的问题。

## 1.3 解题思路

本项目采用多种方法求解硬币中的组合数，并比较各种方法的优劣，以及组合数学中母函数引入的重要意义。

最简单的是用C语言实现暴力穷举法，把所有的可能情况模拟一遍，可以得到三种货币中硬币构成各自货币中1分~10元的全部组合数。但这样的缺点也很明显：首先，效率很低，由于每种货币的硬币种类都很多，所有穷举起来的计算量很大（事实上金额达到190分时计算时间已经很长了）；其次，不能直观的显示各种货币的表现能力。

第二种方法使用C++，通过动态规划的方法逐步求解指定硬币种类和金额的组合数。优点是效率很高；缺点也是不能通过图形直观的显示各种货币的表现能力。

第三种方法是将各个货币的各个面额的硬币的母函数写出来，然后写出该货币的母函数，通过母函数各项的系数理论上可以得出任意金额的组合数。该项目使用MATLAB的符号运算求解高阶多项式。

# 2 算法设计

## 2.1 穷举法

C语言实现

### 2.1.1 穷举法核心思想

举人民币为例，总共有1分、2分、5分、1角、5角、1元共六种硬币，而每一种硬币对应的取值情况可能是0~amout/金额 个。

因此可以使用6个变量，分别表示六种硬币种类，当硬币数量\*硬币金额的累加和等于amount的时候则令计数变量递增一。

如此循环即可计算出六种硬币表示1000分的组合数。

### 2.1.2 穷举法核心代码

int Combination\_Yuan(int amount)

{

int cnt = 0;

for (int a1 = 0; a1 <= amount / 1; ++a1)

{

for (int a2 = 0; a2 <= amount / 2; ++a2)

{

for (int a3 = 0; a3 <= amount / 5; ++a3)

{

for (int a4 = 0; a4 <= amount / 10; ++a4)

{

for (int a5 = 0; a5 <= amount / 50; ++a5)

{

for (int a6 = 0; a6 <= amount / 100; ++a6)

{

if (a1 \* 1 + a2 \* 2 + a3 \* 5 + a4 \* 10 + a5 \* 50 + a6 \* 100 == amount)

{

++cnt;

}

}

}

}

}

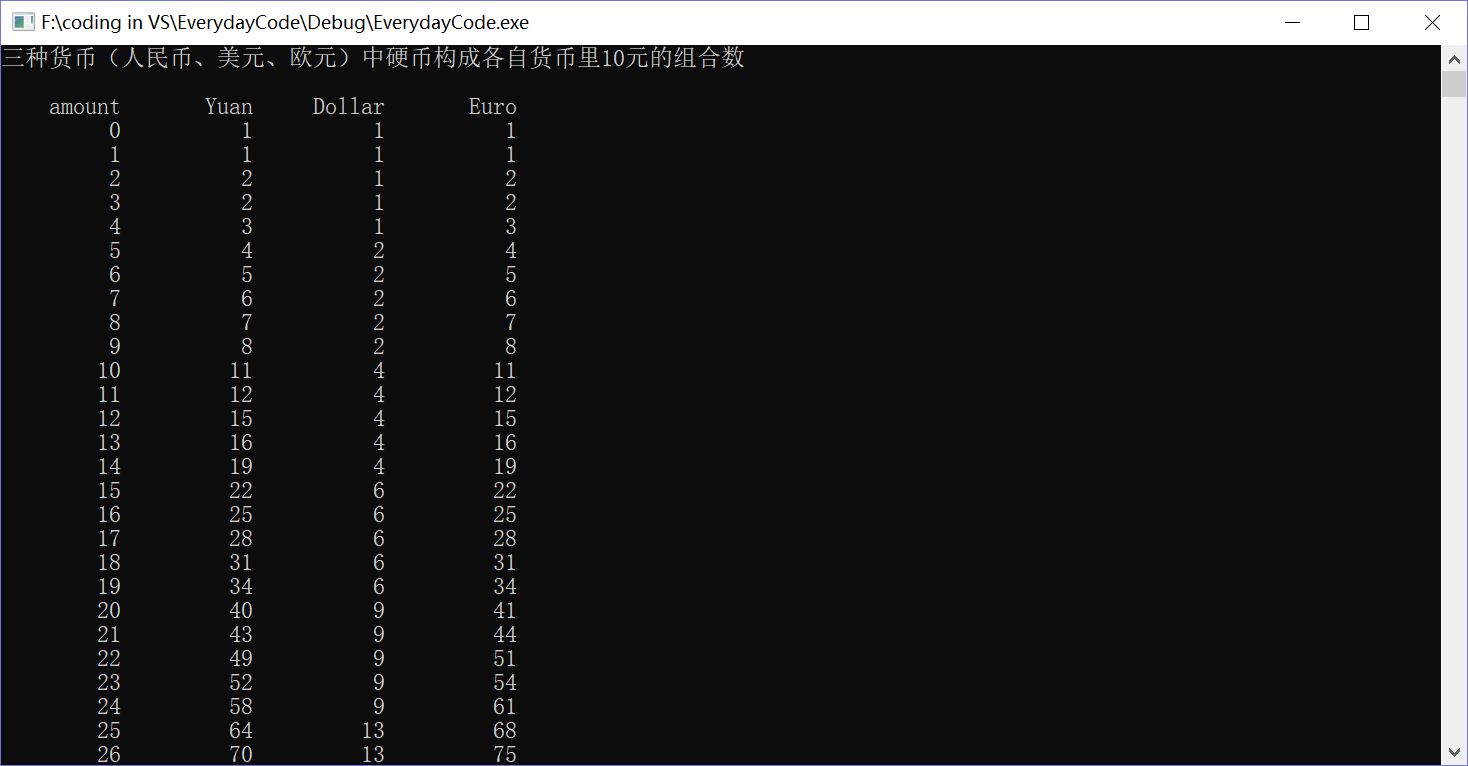
}

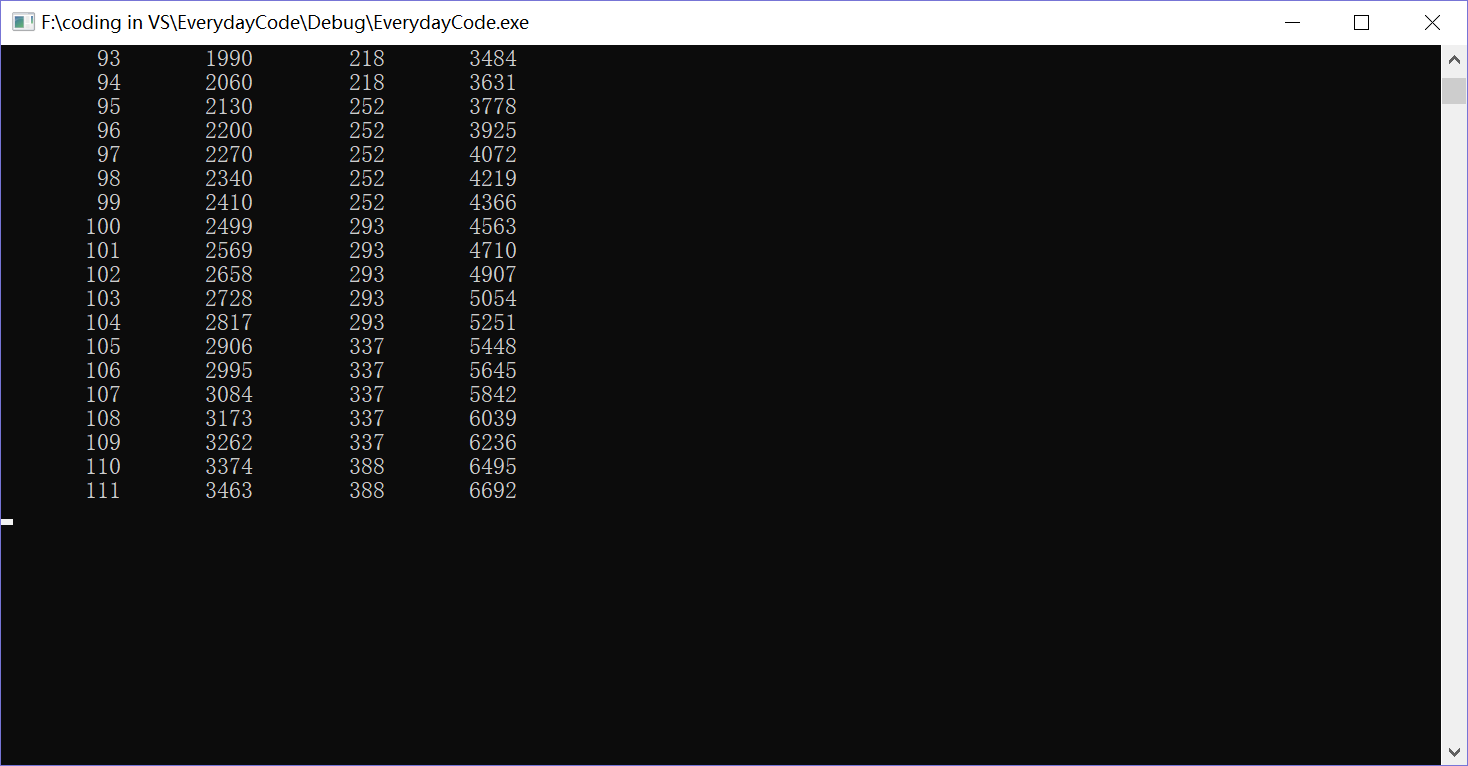
}

return cnt;

}

### 2.1.3 穷举法运行截图





### 2.1.4 穷举法优缺点分析

优点在于思想简单，代码易于编写，因为计算机的计算速度远大于人的手算速度，在金额不大的情况下不失为首选的方法。

然而该算法的时间复杂度达到，当总金额较大的时候对于计算机而言仍是不可解的。本题而言，在我的笔记本上跑到190的时候已经非常非常慢了。

## 2.2 动态规划法

C++实现

### 2.2.1 动态规划法核心思想

我们有m种不同的硬币{v1,v2,…,vm}，要组合成给定的金额sum

即sum=x1\*v1 + x2\*v2 +…+ xm\*vm中x1~xm所有可能的个数

对于第m种硬币，它可以使用0,1,2,…,k次(k=sum/vm)

则对于第m种硬币的不同取值情况我们可以把sum分解成

sum=x1\*v1 + x2\*v2 +…+ 0\*vm

sum=x1\*v1 + x2\*v2 +…+ 1\*vm

…

sum=x1\*v1 + x2\*v2 +…+ k\*vm

定义一个二维数组result[i][sum]表示用前i中硬币构成sum的所有组合数

对于人民币，问题的规模就是i=6，sum=1000的组合数

而在前面连立的等式中xn=0时，相当于result[i-1][sum]种；xn=1时，相当于result[i-1][sum-vm]种；xn=2时，相当于result[i-1][sum – 2\*vm]种。

可以通过第i-1行求出第i行的各项的值

初始条件为任何种类的硬币组成0元钱只有一种情况即每一个硬币都不选，即result[i][0]=1；以及0种硬币构成任意金额没有对应的可能性，即result[0][sum]=0

### 2.2.2 动态规划法核心代码

vector<int> Yuan = { 1,2,5,10,50,100 };

vector<vector<int> >ComYuan(Yuan.size()+1, vector<int>(amount+1, 0)); //增加0种硬币的情况 和 0元钱的情况

void Combination(const vector<int> &coins, vector<vector<int> > &result, const int amount=1000)

{

//把0种硬币组成任意金额的位置置为0

for (int sum = 0; sum <= amount; ++sum)

{ result[0][sum] = 0; }

//任意硬币数量组成0元有1种可能

for (int i = 0; i <= coins.size(); ++i)

{

result[i][0] = 1;

}

for (int i = 1; i <= coins.size(); ++i)

{

for (int sum = 1; sum <= amount; ++sum)

{

for (int k = 0; k <= sum / coins[i-1]; ++k)

{

result[i][sum] += result[i - 1][sum - k \* coins[i-1]];

//conins的i-1位置标识着当前研究的硬币总种类的最后一种

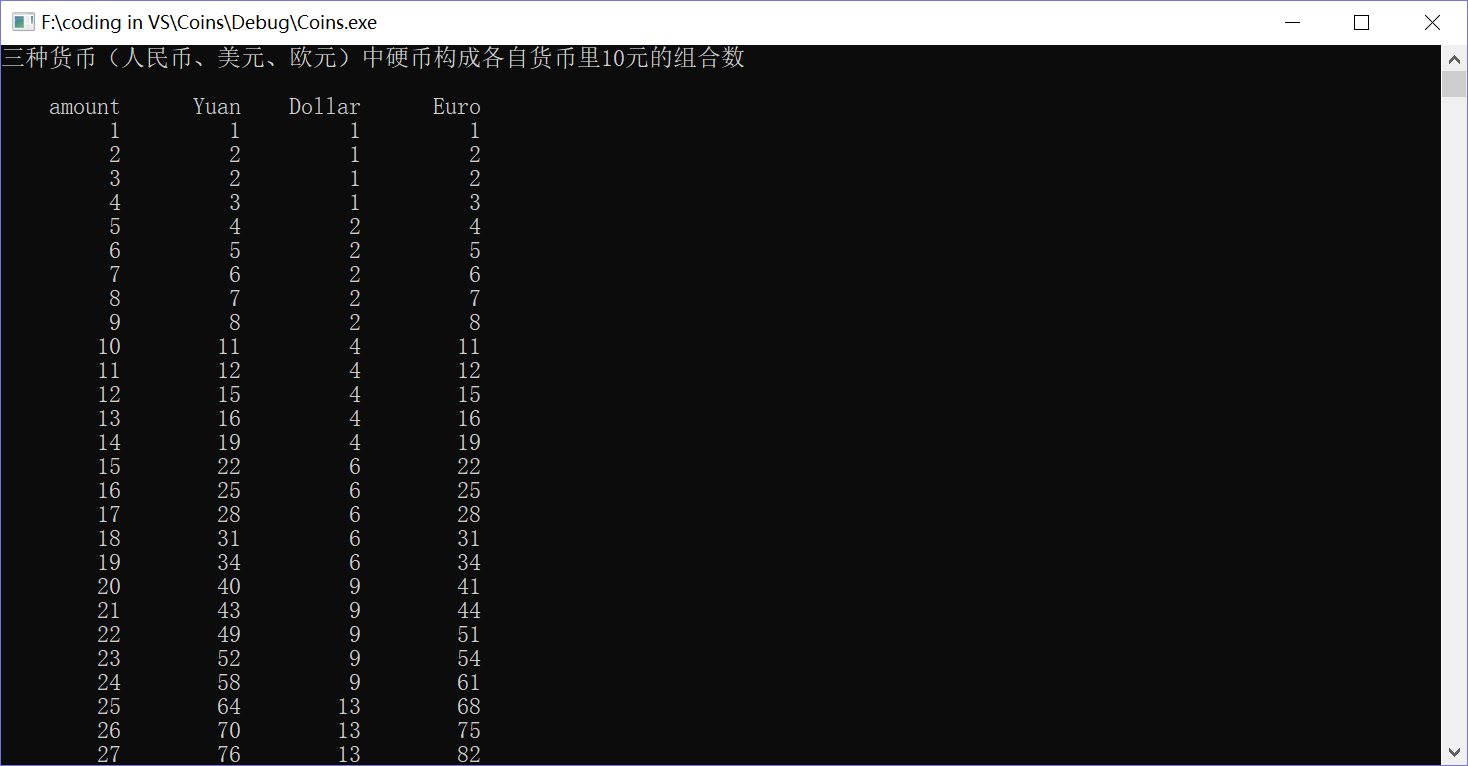
}

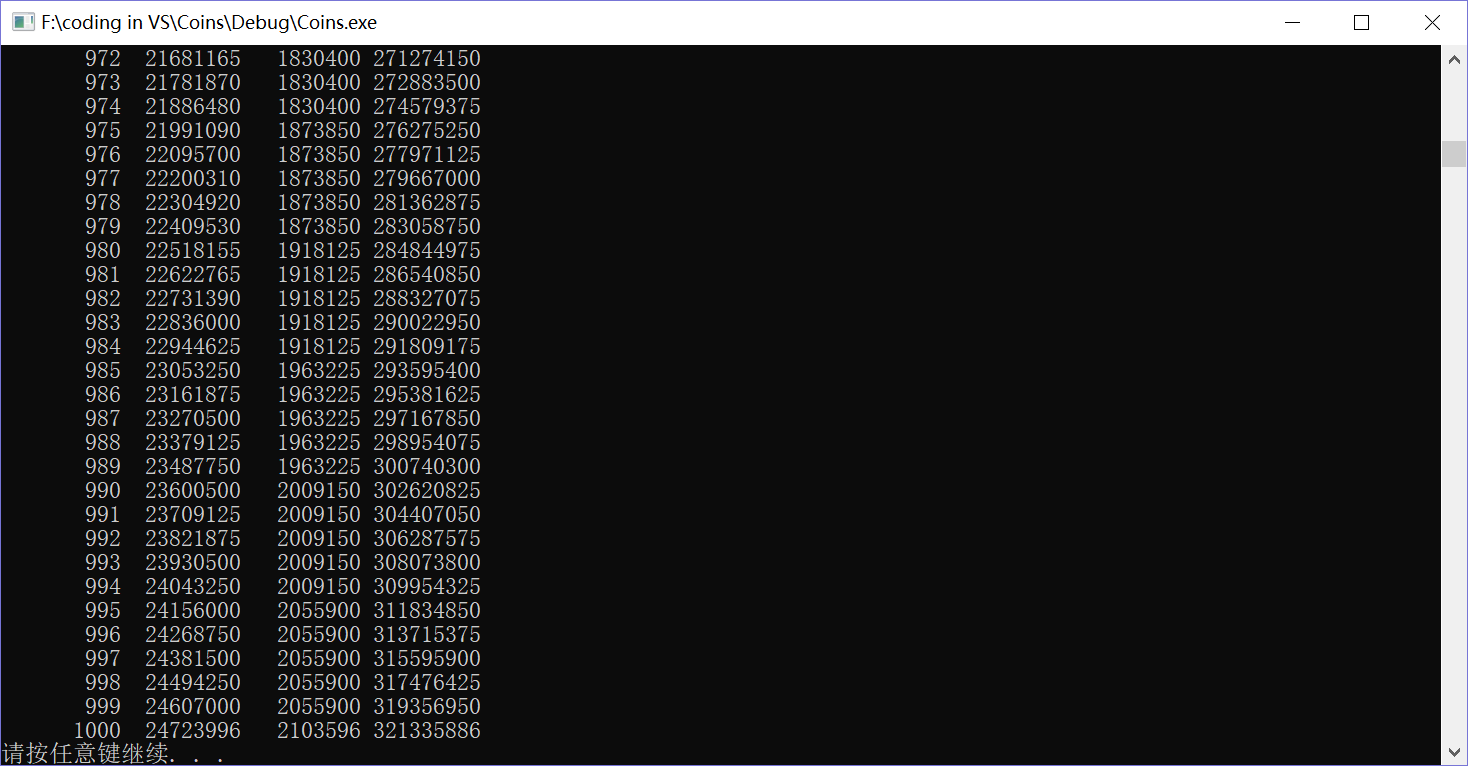
}

}

}

### 2.2.3 动态规划法运行截图





### 2.2.4 动态规划法优缺点分析

优点是动态规划的方法通过问题规模较小时较容易求出的情况递推的找到问题规模较大时的解，类似于课上学的递推关系。尤其与斐波那契数列那一节的01字符串问题很相似，都是根据最后一位的不同取值（取0或者1）限制或者递推的找到前面对应的情况。计算效率很高，可以在很短的时间内计算出1~1000分的所有组合数。

但是缺点是，通过C++编程只能显示求得的组合数的数值，不能绘制函数或表格较为清晰的显现3种货币的组合数差异和表现能力。

## 2.3 母函数法

MATLAB实现

### 3.2.1 母函数法核心思想

根据2-8整数的拆分一节中举出的例子

1分的母函数表示为G(x) = 1 + x + x2 + x3 + x4 + …

可以改写成 G(x) = 1/(1-x)

2分的母函数表示为G(x) = 1 + x2 + x4 + x6 + x8 + …

可以改写成 G(x) = 1/(1-x2)

5分的母函数表示为G(x) = 1 + x5 + x10 + x15 + x20 + …

可以改写成 G(x) = 1/(1-x5)

1角的母函数表示为G(x) = 1 + x10 + x20 + x30 + x40 + …

可以改写成 G(x) = 1/(1-x10)

5角的母函数表示为G(x) = 1 + x50 + x100 + x150 + x200 + …

可以改写成 G(x) = 1/(1-x50)

1元的母函数表示为G(x) = 1 + x100 + x200 + x300 + x400 + …

可以改写成 G(x) = 1/(1-x100)

把它们乘在一起即得到人民币的母函数，通过MATLAB的符号计算方法，可以把高阶多项式展开，并取它的2~1001项，每一项的系数即对应了指数项金额的组合数。

再通过MATLAB的绘图函数绘制曲线图和柱状图直观的体现各个货币的表现能力。

### 3.2.2 母函数法核心代码

*syms x %定义符号变量x*

*%人民币*

*Yuan1=1/(1-x);*

*Yuan1=series(Yuan1,x,'Order',1001); %一分 -> 母函数*

*Yuan2=1/(1-x^2);*

*Yuan2=series(Yuan2,x,'Order',1001); %两分 -> 母函数*

*Yuan3=1/(1-x^5);*

*Yuan3=series(Yuan3,x,'Order',1001); %五分 -> 母函数*

*Yuan4=1/(1-x^10);*

*Yuan4=series(Yuan4,x,'Order',1001); %一角 -> 母函数*

*Yuan5=1/(1-x^50);*

*Yuan5=series(Yuan5,x,'Order',1001); %两角 -> 母函数*

*Yuan6=1/(1-x^100);*

*Yuan6=series(Yuan6,x,'Order',1001); %一元 -> 母函数*

*Yuan=Yuan1\*Yuan2\*Yuan3\*Yuan4\*Yuan5\*Yuan6; %总金额的母函数*

*Yuan=expand(Yuan); %展开多项式*

*Yuan=coeffs(Yuan); %展成标准形式*

*Yuan=Yuan(2:1001) %取前1000位*

*%第一位是总金额等于0的时候, 没有讨论的价值*

*%绘制函数图像*

*Yuanx=1:1:1000;*

*plot(Yuanx,Yuan)*

*hold on;*

*Dollarx=1:1000;*

*plot(Dollarx,Dollar)*

*Eurox=1:1000;*

*plot(Eurox,Euro)*

*legend('人民币','美元','欧元');*

*xlabel('分');*

*ylabel('组合数');*

*figure*

*%绘制柱状图*

*Combine=zeros(3,10);*

*for row=100:100:1000*

*Combine(1,row/100)=Yuan(row);*

*Combine(2,row/100)=Dollar(row);*

*Combine(3,row/100)=Euro(row);*

*end*

*Combine=Combine';*

*bar(Combine)*

*set(gca,'ygrid','on'); %只显示网格的横线*

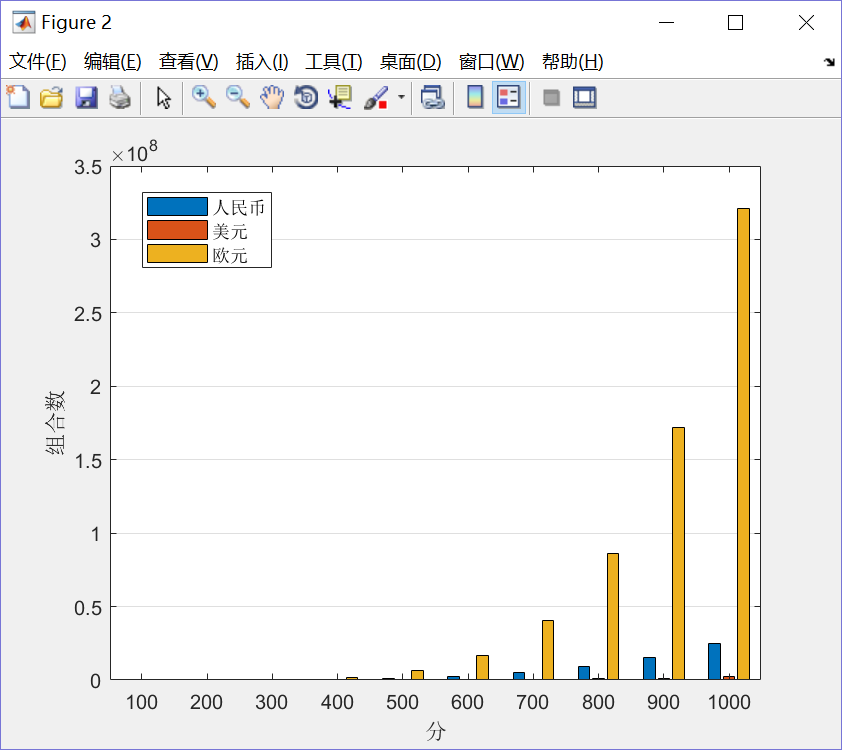
*set(gca,'xticklabel',{'100','200','300','400','500','600','700','800','900','1000'});*

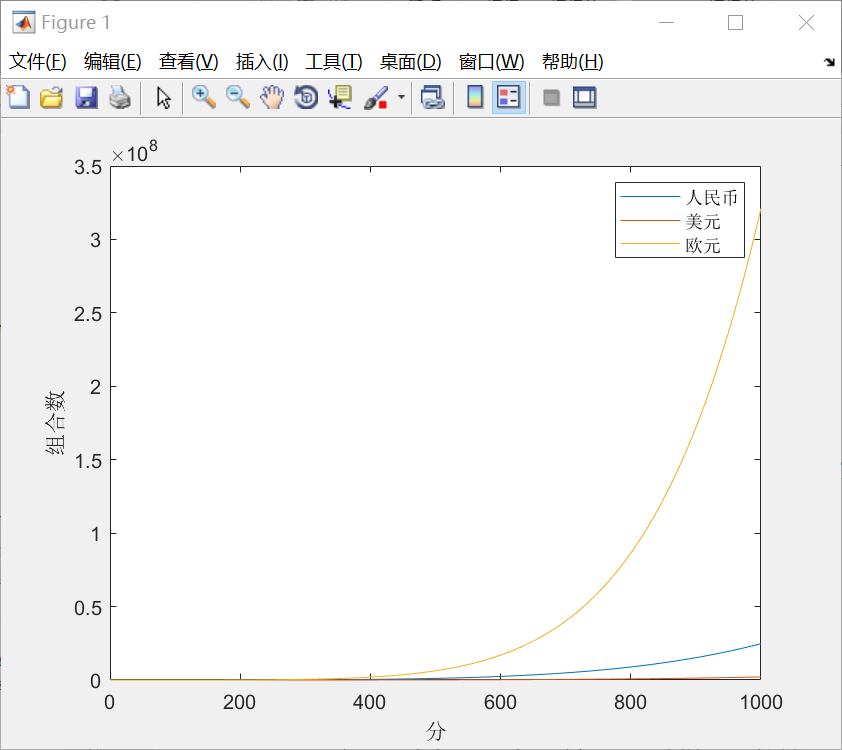
*legend('人民币','美元','欧元');*

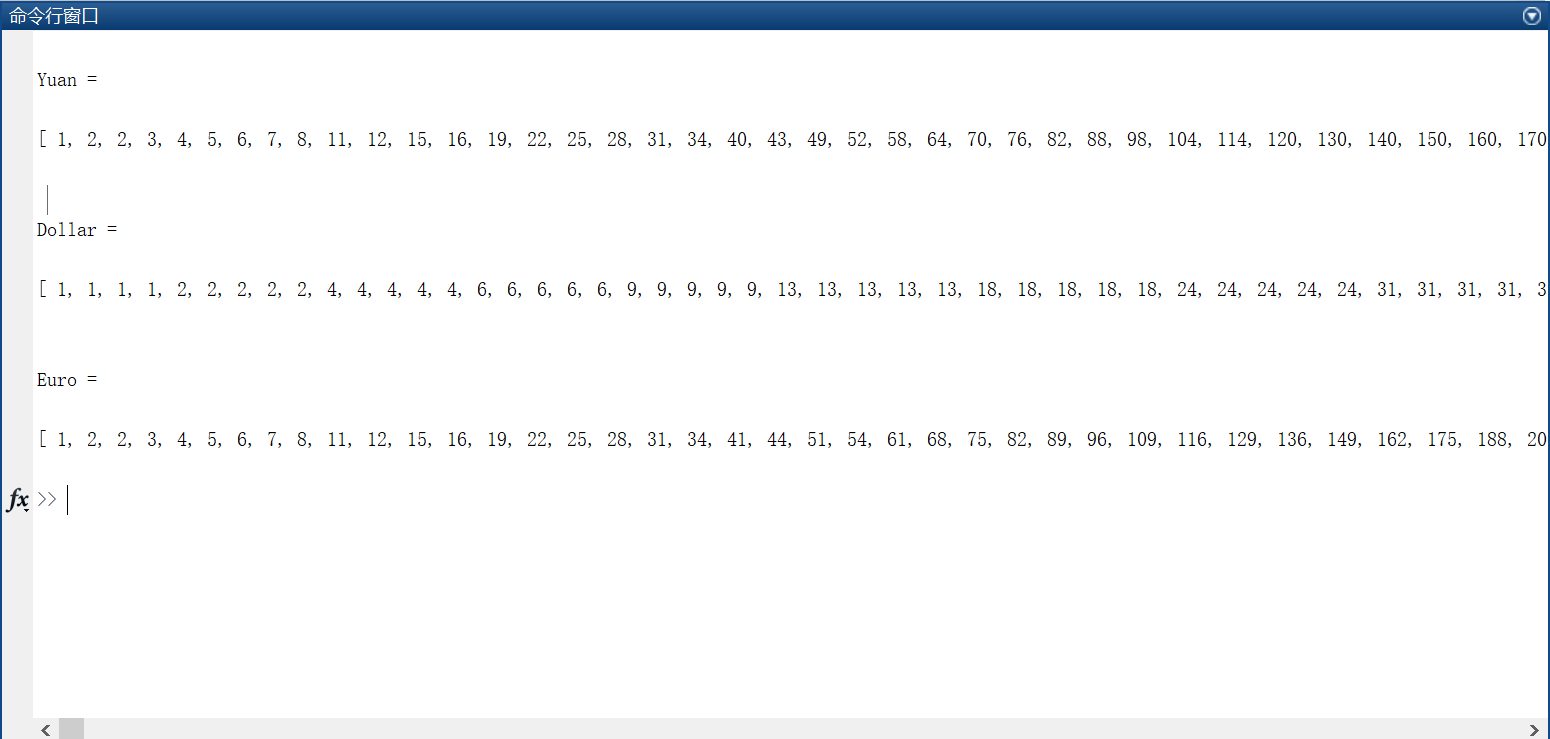
*xlabel('分');*

*ylabel('组合数');*

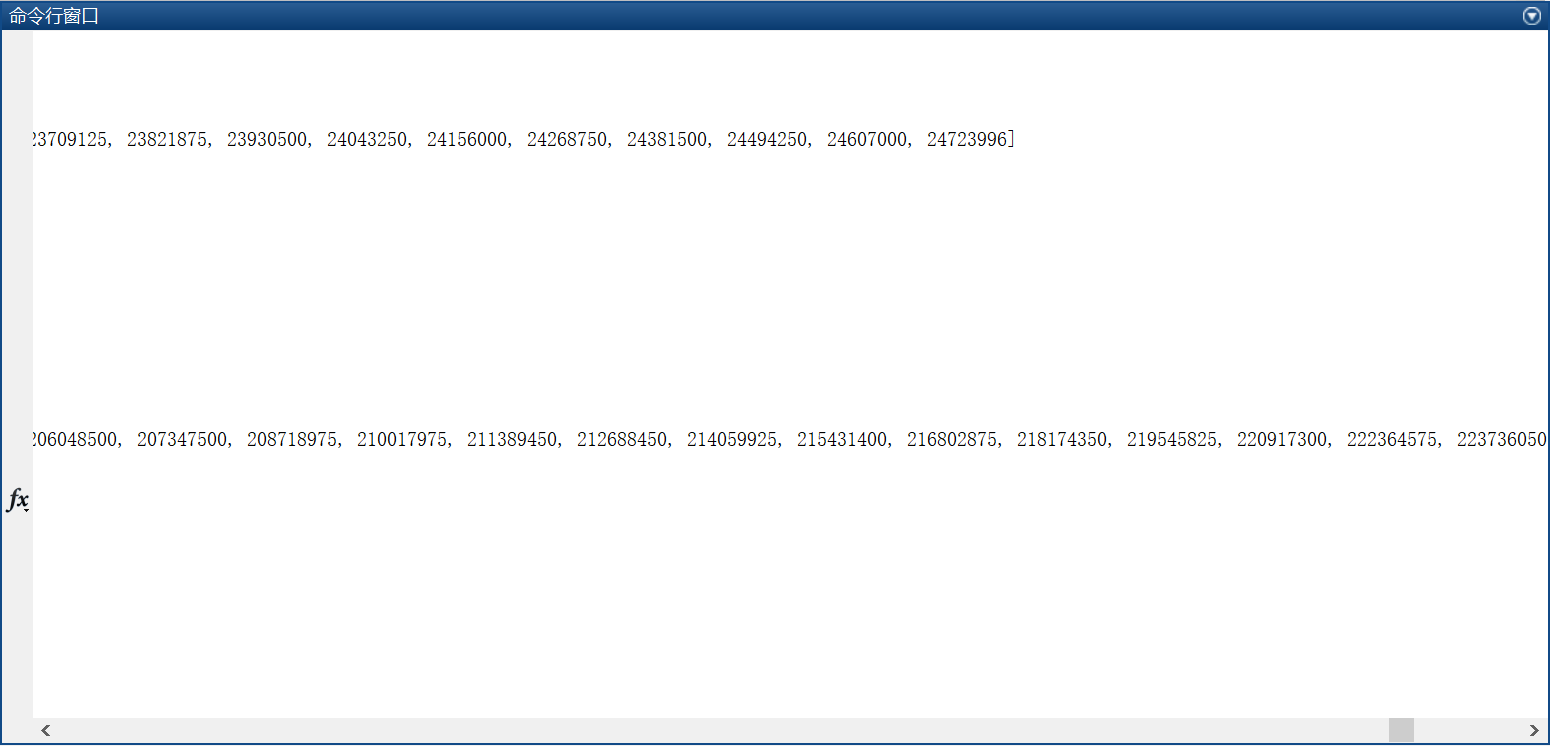
### 3.2.3 母函数法运行截图

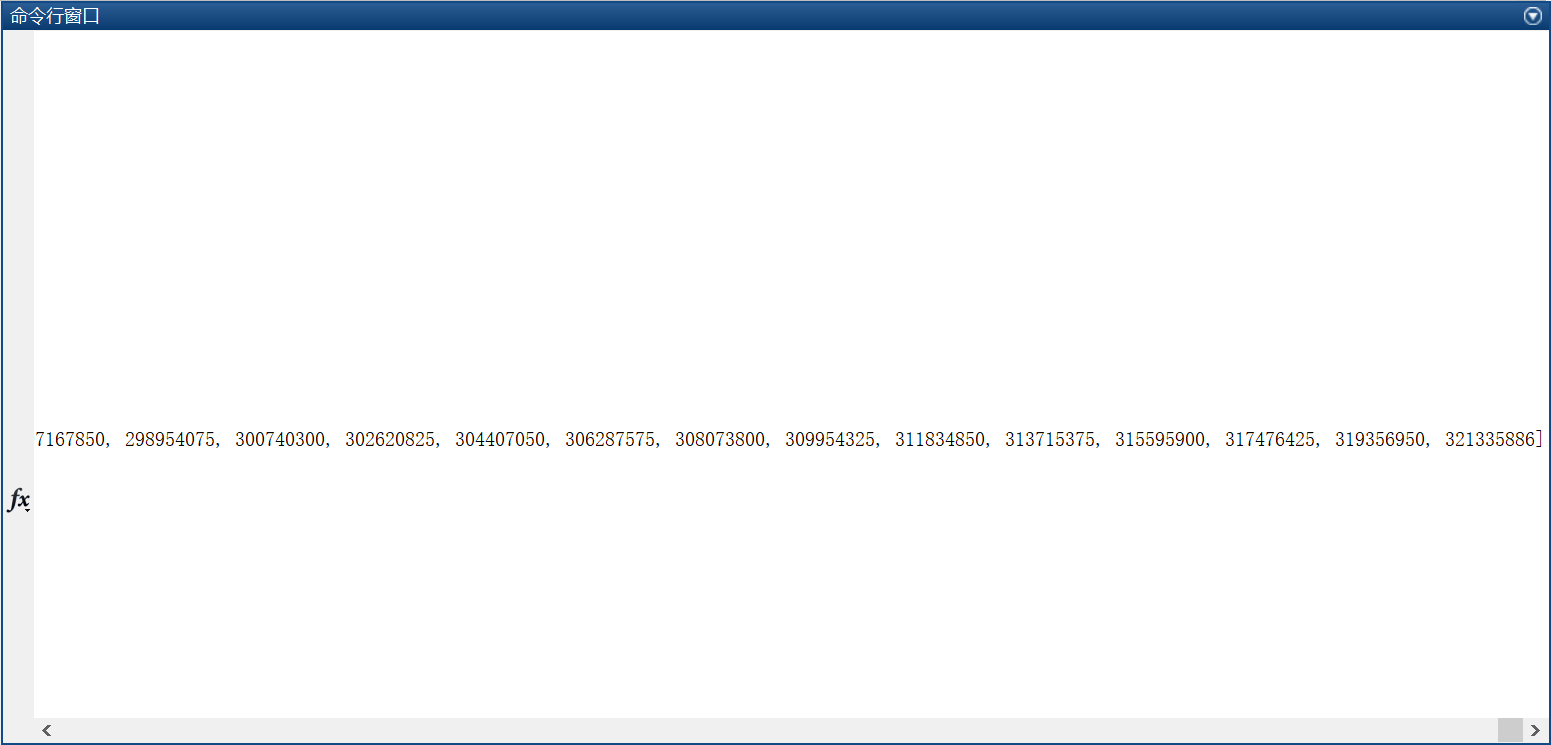












### 3.2.4 母函数法优缺点分析

MATLAB的符号计算及绘图配合母函数的方法，使得该项目的解变得非常直观和清楚；同时由于是解方程，理论上可以同时求出组成任意金额的组合数，非常便捷。

缺点主要在于MATLAB的符号计算的速度较C++还是有一定差距，这里只进行高阶多项式的展开还是可以的，如果函数形式过于复杂可能计算起来效率也比较低。

# 3 结果分析

在表示金额不大的情况下，人民币和欧元的表现能力相仿，美元较为逊色；但当金额增加到最后的1000分时，欧元的表现能力较人民币大了一个数量级，较美元大了两个数量级，差距悬殊。

总体分布上看，美元与正比例函数较为贴合，人民币与nlogn函数较为贴合，而欧元近似成指数增长。

同时美元较其他两种货币比有一个较为有意思的特点，相近的金额的组合数会保持相同，然后呈现阶梯状上升，例如金额在95~109区间时，组合数为252 252 252 252 252 293 293 293 293 293 337 337 337 337 337 337，仔细观察后发现每一个组合数数字都会重复5次，然后阶梯上升到下一个组合数再重复5次，观察了1000个样本点都不加证明的满足这个规律（但是不知道原因，希望老师能够解答），猜想可能是因为美元没有1到5分的过渡硬币，使得相差5美分之内的表现能力较差；而其他两种货币除最开始金额太小的时候都是保持严格单调递增的。

欧元明显比其他两个货币的表现能力强主要归功于欧元中有8种硬币；而美元和人民币虽然都是6种硬币但是表现能力相差一个数量级主要在于美元中较小的硬币种类太少，人民币中第二小的是2分，而美元是5分，使表现能力大大下降。

# 4 结论

三种货币（人民币、美元、欧元）构成各自货币里10元的组合数分别为（人民币24723996，美元2103596，欧元321335886），欧元从表现能力上看比人民币高一个数量级，比美元高两个数量级，表现能力出色；同时美元呈现阶梯上升状态，每个组合数会重复5次然后阶梯上升到下一个阶梯。