

母函数

定义

- 又称生成函数，用多个独立多项式相乘来解决组合问题。
- 只关心多项式的系数，而不关心变量 x 的取值

普通型母函数

HDU 1085

- 题意：给你1,2,5这几个硬币，每一个有a,b,c个，问你最小的不能达到的价值是多少？
- 代码

```
#include<iostream>
#include<cstring>
using namespace std;
const int maxn = 1e4;
int a,b,c;//1 2 5硬币的数量
int t1[maxn]; //中间结果存放系数
int t2[maxn]; //中间结果 存放
int r[maxn]; //最终结果存放
int main(){
while(cin>>a>>b>>c){
    if(a + b + c == 0) break;
    int limit = a + 2*b + 5*c; //最多能表示的钱
    memset(r, 0, sizeof r);
    for(int i=0; i<=limit+1; i++){
        t1[i] = t2[i] = 1;
    }
    for(int i=0; i<=a; i++){ //枚举第一个多项式的次数 i
        for(int j=0; j<=2*b; j+=2){ //枚举第二个多项式的次数 j
            r[i+j] += t1[i]*t2[j]; //得到的是对i+j次数的贡献
        }
    }
    for(int i=0; i<=a+2*b; i++){ //多项式1和多项式2相乘的中间结果最高次数不超过
a+2b
        t1[i] = r[i]; //t1暂存多项式1和多项式2相乘的中间结果 t2现在代表多项式3的系
数
        r[i] = 0;
    }

    for(int i=0; i<=a+2*b; i++){
        for(int j=0; j<=5*c; j+=5){
            r[i+j] += t1[i]*t2[j];
        }
    }

    for(int i=0; i<=limit+1; i++){
```

```

        if(r[i] == 0){
            cout<<i<<endl;
            break;
        }
    }
}
return 0;
}

```

HDU 1171

- 题意：题目大意有n个物品，每个物品都有价值v和数量m，要分成两堆，使价值尽量接近
- 解法1：当成多重背包做，先转化为0-1背包，然后把背包容量定义成总价值的一般，并且定义每件物品的价值和重量一样，然后解0-1背包问题
- 解法二：用母函数做。相当于各种物品生成的多项式独立，最后乘起来可以得到各种组合价值的方案数量，从sum/2往后找第一个系数不为0的项即可

```

#include<iostream>
#include<cstring>
using namespace std;
const int maxn = 1e6;
const int N = 1010;
int t1[maxn]; //中间结果存放系数
int t2[maxn]; //中间结果 存放
int r[maxn]; //最终结果存放
int n;
int v[N], num[N];
int main(){
    ios::sync_with_stdio(false);
    while(cin>>n && n >= 0){
        if(n == 0) cout<<"0 0"<<endl;
        int sum = 0;
        for(int i=1; i<=n; i++){
            cin>>v[i]>>num[i];
            sum += v[i]*num[i];
        }

        for(int i=0; i<=sum; i++){
            t1[i] = 0;
            t2[i] = 0;
        }
        //第一个多项式
        for(int i=0; i<=v[1]*num[1]; i+=v[1]){
            t1[i] = 1;
        }
        //最高次记录
        int tot = v[1]*num[1];

        for(int i=2; i<=n; i++){ //逐步乘多项式
            for(int j=0; j<=tot; j++){ //累积多项式
                for(int k=0; k<=v[i]*num[i]; k+=v[i]){ //当前多项式

```

```

                t2[j+k] += t1[j];
            }
        }
        //重新布局
        tot += v[i]*num[i];
        for(int i=0; i<=tot; i++){
            t1[i] = t2[i]; //存到中间结果
            t2[i] = 0;
        }
    }
    int half = (tot+1) / 2;
    for(int i=half; i<=tot; i++){
        if(t1[i] != 0){
            cout<<i<<" "<<sum-i<<endl;
            break;
        }
    }
}
return 0;
}

```

HDU 1709 放砝码

- 题意：给定几种重量的砝码，每种砝码只有一个，砝码可以放在天平左右两边，问不能称出来的重量有哪些？
- 思路：分两种情况讨论，放在一边，放在两边，只要进行加减就可以了，减的时候取绝对值。

```

#include<iostream>
#include<cstring>
#include<cmath>
using namespace std;
const int maxn = 1e4+100;
const int N = 110;
int t1[maxn]; //中间结果存放系数
int t2[maxn]; //中间结果 存放
int ans[maxn];
int n,m;
int w[N];
int main(){
    ios::sync_with_stdio(false);
    while(cin>>n){
        int sum = 0;
        for(int i=1; i<=n; i++){
            cin>>w[i];
            sum += w[i];
        }
        for(int i=0; i<=sum; i++){
            t1[i] = t2[i] = 0;
        }
        t1[0] = t1[w[1]] = 1;
    }
}

```

```

    for(int i=2; i<=n; i++){ //第i个多项式 枚举第i个砝码的情况
        for(int j=0; j<=sum; j++){ //前一个中间结果
            //for(int k=0; k<=w[i]; k++){
            //两个砝码放在一边
            t2[j] += t1[j]; //k=0
            t2[j+w[i]] += t1[j]; //k=w[i]

            //两个砝码分别放在两边
            t2[abs(j-w[i])] += t1[j];
        }
        for(int j=0; j<=sum; j++){
            t1[j] = t2[j];
            t2[j] = 0;
        }
    }
    int cnt = 0;

    for(int i=1; i<=sum; i++){
        if(t1[i] != 0) ans[cnt++] = i;
    }
    cout<<cnt<<endl;
    if(cnt != 0){
        cout<<ans[0];
        for(int i=1; i<cnt; i++) cout<<" "<<ans[i];
        cout<<endl;
    }

}
return 0;
}

```

HDU 2096

- 题意：有几种硬币，可以使用最多100枚硬币组成价值N，问有多少种组成方案
- 思路1：二维母函数，第一维代表价值，第二维代表用去的硬币数量，初始化t[0][0] = 1
- 思路2：用动态规划做，dp[i][j]表示组成价值i用了j枚硬币，显然这个状态可以由dp[v][j-1]再加上一枚(i-v)的硬币转移过来

```

#include<iostream>
#include<cstring>
#include<cmath>
using namespace std;
const int maxn = 1e3+100;
const int N = 110;
int t1[maxn][105]; //中间结果存放系数
int t2[maxn][105]; //中间结果 存放
int n,m;

```

```

//母函数做法
int a[5] = {1, 5, 10, 25, 50};
int main(){
    memset(t1, 0, sizeof t1);
    memset(t2, 0, sizeof t2);
    t1[0][0] = 1; //预设一个硬币0的多项式 只含有1
    for(int i=0; i<5; i++){ //处理剩下的硬币的多项式
        for(int j=0; j<251; j++){ //中间结果
            for(int k=0; k+j<251; k+=a[i]){ //枚举当前硬币多项式
                for(int l=0; l+k/a[i]<=100; l++){ //枚举之前已经用了的硬币数量
                    //当前不能超过100
                    t2[j+k][l+k/a[i]] += t1[j][l];
                }
            }
        }
        for(int j=0; j<251; j++){
            for(int k=0; k<=100; k++){
                t1[j][k] = t2[j][k];
                t2[j][k] = 0;
            }
        }
    }

    while(cin>>n){
        int ans = 0;
        for(int i=0; i<=100; i++) ans += t1[n][i];
        cout<<ans<<endl;
    }
}

// DP做法
//int a[5] = {1, 5, 10, 25, 50};
//int dp[maxn][105];
//int main(){
//    ios::sync_with_stdio(false);
//    while(cin>>n){
//        memset(dp, 0, sizeof dp);
//        dp[0][0] = 1;
//        for(int i=0; i<5; i++){
//            for(int j=1; j<=100; j++){
//                for(int k=a[i]; k<=n; k++){
//                    dp[k][j] += dp[k-a[i]][j-1];
//                }
//            }
//        }
//        int ans = 0;
//        for(int i=0; i<=100; i++) ans += dp[n][i];
//        cout<<ans<<endl;
//    }
//    return 0;
//}

```

指数型母函数

- 定义：从多重集 $M = \{\infty a_1, \infty a_2, \infty a_3, \dots, \infty a_n\}$ 中选出 K 个元素的 K -排列中，若限定元素 a_i 出现的次数集合为 M_i ，则该组合数序列的母函数为： $x^m/m!$ ， $m \in M_i$ 的所有方法之和再连乘。

HDU 1521

- 题意：从 n 种物品中选取一共 m 个物品，问一共有多少种排列数？

```
#include<iostream>
#include<cstring>
using namespace std;

typedef long long ll;

double fac[15];
int n,m;
int num[15];
double t1[15],t2[15];
void init(){
    fac[0] = 1;
    for(int i=1; i<=12; i++){
        fac[i] = fac[i-1] * i;
    }
}

int main(){
    init();
    while(scanf("%d %d", &n, &m) != EOF){
        for(int i=1; i<=n; i++) scanf("%d", &num[i]);
        memset(t1, 0, sizeof t1);
        memset(t2, 0, sizeof t2);
        for(int i=0; i<=num[1]; i++) t1[i] = 1/fac[i];
        for(int i=2; i<=n; i++){ //剩余n-1个多项式
            for(int j=0; j<=m; j++){ //中间结果多项式枚举项
                for(int k=0; k<=num[i]&&j+k<=m; k++){ //当前多项式枚举项
                    t2[j+k] += t1[j]/fac[k];
                }
            }
            for(int j=0; j<=m; j++){
                t1[j] = t2[j];
                t2[j] = 0;
            }
        }
        printf("%.0f\n", t1[m]*fac[m]);
    }
}
```

HDU 2065

- 题意：用ABCD四个字符组成长度为N的字符串，其中AC只能不出现或者出现偶数次，问有多少种组法
- 思路：指数型母函数加泰勒公式合并加重新展开

```
#include<iostream>
using namespace std;

typedef long long ll;

ll qpow(ll a, ll n, ll mod){
    ll ans = 1;
    while(n){
        if(n & 1)
            ans = (ans * a) % mod;
        n >>= 1;
        a = (a * a) % mod ;
    }
    return ans;
}

int main(){
    ll n,t;
    int cas = 1;
    while(cin >> t){
        if(t == 0) break;
        while(t--){
            cin>>n;
            ll ans = ( qpow(4 , n-1, 100LL) + qpow(2 , n-1, 100LL) ) % 100;
            cout<<"Case "<<cas++<<": "<<ans<<endl;
        }
        cas = 1;
        cout<<endl;
    }
}
```