第8章　算法设计初体验

第86课　记录结果再利用

《信息学奥赛一本通·编程启蒙 C++版》

如果说“穷举法”充分利用了计算机快速计算的能力，那么“递推法”、“记忆化搜索”、“动态规划”等这些大名鼎鼎的算法都是充分利用了计算机的记忆能力。

【例 86.1】 上台阶

【题目描述】

楼梯有 n(71>n>0)阶台阶,上楼时可以一步上 1 阶,也可以一步上 2 阶,也可以一步上 3 阶，编程计算共有多少种不同的走法。

【输入格式】

输入的每一行包括一组测试数据，即为台阶数 n。最后一行为 0，表示测试结束。

【输出格式】

每一行输出对应一行输入的结果，即为走法的数目。

【样例输入】

```
1
2
3
4
0
```

【样例输出】

```
1
2
4
7
```

【代码实现 1】

```cpp
#include <bits/stdc++.h>
using namespace std;
long long x,a[75];
int main(){
    a[1]=1;
    a[2]=2;
    a[3]=4;
    for(int i=4;i<=75;i++)
        a[i]=a[i-1]+a[i-2]+a[i-3];
   while(cin>>x){
        if(x==0) break;
        cout<<a[x]<<endl;
   }
   return 0;
}
```

【代码实现 2】

```cpp
#include <bits/stdc++.h>
using namespace std;
long long x,a[75];
long long f(int x){
    if(x==1) return 1;
    if(x==2) return 2;
    if(x==3) return 4;
    if(a[x]>0) return a[x];
    a[x]=f(x-1)+f(x-2)+f(x-3);
    return a[x];
}
int main(){
   while(cin>>x){
       if(x==0) break;
       cout<<f(x)<<endl;
   }
   return 0;
}
```

【例 86.2】 01 背包问题

【题目描述】

一个旅行者有一个最多能装 M 公斤的背包，现在有 n 件物品，它们的重量分别是 W_1，W_2，...,W_n.它们的价值分别为 C_1,C_2,... ,C_n，求旅行者能获得最大总价值。

【输入格式】

第一行:两个整数，M(背包容量，M<=200)和 N(物品数量，N<= 30);第 2..N＋1 行:每行二个整数 W，C，表示每个物品的重量和价值。

【输出格式】

仅一行，一个数，表示最大总价值。

【样例输入】

```
10 4
2 1
3 3
4 5
7 9
```

【样例输出】

```
12
```

【代码实现 1】

```
#include<bits/stdc++.h>
using namespace std;
#define MAX 1001
int n,weight;
int w[MAX],v[MAX];
int dp[MAX][MAX];
int f(int n,int weight){
   if(n==1) {
      if(weight>=w[1])  return v[1];
      else return 0;
   }
   if(dp[n][weight]>0) return dp[n][weight];
   if(weight<w[n]) dp[n][weight]=f(n-1,weight);
     else
       dp[n][weight]=max(f(n-1,weight),f(n-1,weight-w[n])
  +v[n]);
   return   dp[n][weight];
}
int main()
{
    scanf("%d%d",&weight,&n);
    for(int i=1;i<=n;i++) scanf("%d%d",&w[i],&v[i]);
    memset(dp,0,sizeof(dp));
    printf("%d\n",f(n,weight));
    return 0;
}
```

【代码实现 2】

```cpp
#include<bits/stdc++.h>
using namespace std;
#define MAX 1001
int n,weight;
int w[MAX],v[MAX];
int dp[MAX][MAX];
int main()
{
  scanf("%d%d",&weight,&n);
  for(int i=1;i<=n;i++) scanf("%d%d",&w[i],&v[i]);

  memset(dp,0,sizeof(dp));
  for(int i=1;i<=n;i++){
    for(int j=0;j<=weight;j++){
      if(j<w[i]){
        dp[i][j]=dp[i-1][j];
      }
      else{
        dp[i][j]=max(dp[i-1][j],dp[i-1][j-w[i]]+v[i]);
      }
    }
  }
  printf("%d\n",dp[n][weight]);
  return 0;
}
```

【例 86.3】 完全背包问题

【题目描述】

设有 n 种物品，每种物品有一个重量及一个价值。但每种物品的数量是无限的，同时有一个背包，最大载重量为 M，今从 n 种物品中选取若干件(同一种物品可以多次选取)，使其重量的和小于等于 M，而价值的和为最大。

【输入格式】

第一行:两个整数，M(背包容量，M<=200)和 N(物品数量，N<=30);第 2..N＋1 行:每行二个整数 W_i,C_i，表示每个物品的重量和价值。

【输出格式】

仅一行，一个数，表示最大总价值。

【样例输入】

```
10 4
2 1
3 3
4 5
7 9
```

【样例输出】

```
max=12
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
#define MAX 1001
int n,weight;
int w[MAX],v[MAX];
int dp[MAX][MAX];
int f(int n,int weight){
   if(n==1) {
      if(weight>=w[1])  return v[1]*(weight/w[1]);
      else return 0;
   }
   if(dp[n][weight]>0) return dp[n][weight];
   if(weight<w[n]) dp[n][weight]=f(n-1,weight);
      else{
        dp[n][weight]=f(n-1,weight);
        for(int i=1;i*w[n]<=weight;i++)
           dp[n][weight]=max(dp[n][weight],f(n-1,weight-i*w[n])+v[n]*i);
 }
   return   dp[n][weight];
}
int main()
{
    scanf("%d%d",&weight,&n);
    for(int i=1;i<=n;i++) scanf("%d%d",&w[i],&v[i]);
    memset(dp,0,sizeof(dp));
    printf("max=%d\n",f(n,weight));
    return 0;
}
```

【例 86.4】 混合背包

【题目描述】

一个旅行者有一个最多能装 V 公斤的背包，现在有 n 件物品，它们的重量分别是 W_1，W_2，...,W_n，它们的价值分别为 C_1,C_2...C_n。有的物品只可以取一次(01 背包)，有的物品可以取无限次（完全背包)，有的物品可以取的次数有一个上限(多重背包)。求解将哪些物品装入背包可使这些物品的费用总和不超过背包容量且价值总和最大。

【输入格式】

第一行:二个整数，M(背包容量，M<=200)，N(物品数量，N<=30);

第 2..N+1 行:每行三个整数 W_i,C_i,P_i，前两个整数分别表示每个物品的重量，价值，第三个整数若为 0，则说明此物品可以购买无数件，若为其他数字，则为此物品可购买的最多件数(P)。

【输出格式】

仅一行，一个数，表示最大总价值。

【样例输入】

```
10  3
2  1  0
3  3  1
4  5  4
```

【样例输出】

```
11
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
#define MAX 1001
int n,weight;
int w[MAX],v[MAX],p[MAX];
int dp[MAX][MAX];
int f(int n,int weight){
    if(n==1) {
      if(weight>=w[1])
        if(p[1]>weight/w[1]) return v[1]*(weight/w[1]);
         else return v[1]*p[1];
      else return 0;
    }
    if(dp[n][weight]>0) return dp[n][weight];
    if(weight<w[n]) dp[n][weight]=f(n-1,weight);
       else{
          dp[n][weight]=f(n-1,weight);
          for(int i=1;i<=p[n]&&i*w[n]<=weight;i++)
             dp[n][weight]=max(dp[n][weight],f(n-1,
  weight-i*w[n])+v[n]*i);
 }
    return   dp[n][weight];
}
int main()
{
    scanf("%d%d",&weight,&n);
    for(int i=1;i<=n;i++) {
       scanf("%d%d%d",&w[i],&v[i],&p[i]);
       if(p[i]==0) p[i]=weight/w[i]; //把完全背包转换成
  多重背包
    }
    memset(dp,0,sizeof(dp));
    printf("%d",f(n,weight));
    return 0;
}
```

练 86.1 旅行

【题目描述】

你要进行一个行程为 7000KM 的旅行，现在沿途有些汽车旅馆，为了安全起见，每天晚上都不开车，住在汽车旅馆，你手里现在已经有一个旅馆列表，用离起点的距离来标识，如下：

0， 990， 1010， 1970， 2030， 2940， 3060

3930， 4060， 4970， 5030， 5990， 6010， 7000

但在出发之前可能还要增加一些旅馆。

现在旅行社为了节约成本，要求每天至少行驶 A 公里，国家旅行社为了安全起见，要求每天最多只能行驶 B 公里。

你想知道一共有多少种旅行方案。

【输入格式】

第一行输入 A，第二行输入 B，第三行输入 N（0≤N≤20），表示在出发之前又新增 N 个汽车旅馆；接下来 N 行，每行一个整数 m，表示旅馆离起点的距离（0<m<7000）。注意：没有任意两个旅馆在同一位置。

【输出格式】

输出一共有多少种旅行方案。

【样例输入】

500

1500

0

【样例输出】

64

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int a,b,n,ans[40],r[40]={0,990,1010,1970,2030,2940,3060,3930,4060,4970,5030,5990,6010,7000};
int main(){
    scanf("%d %d %d",&a,&b,&n);
    for (int i=14;i<14+n;i++) cin>>r[i];//输入
    sort(r,r+14+n);
    ans[0]=1; //起始点默认一套方案
    for (int i=1;i<14+n;i++){//枚举所有点
        for (int j=0;j<i;j++){//枚举这个点之前的点
            if (r[i]-r[j]>=a&&r[i]-r[j]<=b){
                //如果这两个点之间的距离符合要求
                ans[i]+=ans[j];//这个点可获得前面那个点的所有可能
            }
        }
    }
    cout<<ans[13+n]<<endl;//输出到终点时的所有可能
    return 0;
}
```

练 86.2 采药问题

【题目描述】

辰辰是个很有潜能、天资聪颖的孩子，他的梦想是称为世界上最伟大的医师。为此，他想拜附近最有威望的医师为师。医师为了判断他的资质，给他出了一个难题。医师把他带到个到处都是草药的山洞里对他说：“孩子，这个山洞里有一些不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。我会给你一段时间，在这段时间里，你可以采到一些草药。如果你是一个聪明的孩子，你应该可以让采到的草药的总价值最大。”

如果你是辰辰，你能完成这个任务吗?

【输入格式】

输入的第一行有两个整数 T（1<=T<=1000）和 M（1<=M<=100），T 代表总共能够用来采药的时间，M 代表山洞里的草药的数目。接下来的 M 行每行包括两个在 1 到 100 之间（包括 1 和 100）的的整数，分别表示采摘某株草药的时间和这株草药的价值。

【输出格式】

输出只包括一行，这一行只包含一个整数，表示在规定的时间内，可以采到的草药的最大总价值。

【样例输入】

```
70 3
71 100
69 1
1 2
```

【样例输出】

```
3
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,m;
int sj[1000005],jz[1000005],mm[1005][1005];
int f(int n,int m){
    int y=0;
    if(mm[n][m]>0) return mm[n][m];
    if(n==0 || m==0) return 0;
    int x=f(n-1,m);
    if(m-sj[n]>=0) y=f(n-1,m-sj[n])+jz[n];
    if(x>y){
        mm[n][m]=x;
        return x;
    }
    else{
        mm[n][m]=y;
        return y;
    }
}
int main(){
    cin>>m>>n;
    for(int i=1;i<=n;i++){
        cin>>sj[i]>>jz[i];
    }
    cout<<f(n,m);
    return 0;
}
```

练 86.3 货币系统

【题目描述】

给你一个 n 种面值的货币系统，求组成面值为 m 的货币有多少种方案。

【输入格式】

第一行为 n 和 m。

【输出格式】

一行，方案数。

【样例输入】

```
3 10          //3 种面值组成面值为 10 的方案
1             //面值 1
2             //面值 2
5             //面值 5
```

【样例输出】

```
10            //有 10 种方案
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
long long n,m;
long long a[1000005],mm[10005][10005];
long long f(long long n,long long m){
    long long sum=0;
    if(m==0) return 1;
    if(n==0 && m>0) return 0;
    if(mm[n][m]>0) return mm[n][m];
    for(int i=0;;i++){
        if(m-i*a[n]>=0) sum+=f(n-1,m-i*a[n]);
        else break;
    }
    mm[n][m]=sum;
    return sum;
}
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++){
        cin>>a[i];
    }
    cout<<f(n,m);
    return 0;
}
```

谢谢！