POJ3624

题目描述（POJ3624）：贝西在商场的珠宝店发现一个魅力手镯。她想从 n（1≤n≤3402）个可用的装饰物中选择尽可能好的装饰物去装饰它。每个装饰物都有一个重量 w_i（1≤w_i≤400），以及一个期望值 d_i（1≤d_i≤100），最多可以使用一次。贝西希望装饰物的总重量不超过 m（1≤m≤12880）。给定 n 和 m，并列出装饰物的重量和期望值列表，计算可能的最大期望值之和。

输入：第 1 行包含两个整数 n 和 m。接下来的 n 行，每行都包含两个整数，分别表示装饰物的重量和期望值。

输出：单行输出一个整数，它是在给定权重约束的情况下可以达到的最大期望值的总和。

输入样例	输出样例
4 6 1 4 2 6 3 12 2 7	23

题解：

1. 算法设计

本题为 **01 背包问题**，**可以采用动态规划解决**，**也可以采用回溯法（子集树）解决**，但是**不带优化就会超时，需要剪枝优化。**

约束函数为 **cw+w[i]≤m**，其中 w[i]为第 i 个物品的重量，m 为背包容量。

限界函数为 **cp+brp>bestp**，其中，cp 表示当前装入背包的物品价值，brp 表示剩余容量可容纳的剩余物品的最大价值，bestp 表示当前最优值。

2. 算法实现

```
struct goods{
   int id; //序号
   double d;//单位重量价值
}a[maxn];

bool cmp(goods a,goods b){//按照物品单位重量价值由大到小排序
   return a.d>b.d;
}

double Bound(int i){//当前背包的总价值 cp＋剩余容量可容纳的最大价值
   int cleft=m-cw;//剩余的背包容量
   double brp=cp*1.0;
   while(i<=n&&w[a[i].id]<=cleft){
      cleft-=w[a[i].id];
      brp+=1.0*v[a[i].id];
      i++;
   }
   if(i<=n)
      brp+=cleft*a[i].d;
   return brp;
}
void Backtrack(int t){
   if(t>n){
      bestp=cp;
      return;
   }
   if(cw+w[a[t].id]<=m){//约束
      cw+=w[a[t].id];
      cp+=v[a[t].id];
      Backtrack(t+1);
      cw-=w[a[t].id];
      cp-=v[a[t].id];
   }
   if(Bound(t+1)>1.0*bestp)//限界
     Backtrack(t+1);
}
int main(){
   scanf("%d%d",&n,&m);
   for(int i=1;i<=n;i++)
      scanf("%d%d",&w[i],&v[i]);
   for(int i=1;i<=n;i++){
      a[i].id=i;
      a[i].d=1.0*v[i]/w[i];
   }
   sort(a+1,a+n+1,cmp);
   Backtrack(1);
   printf("%d\n",bestp);
   return 0;
}
```

P2819

题目描述（P2819）：给定无向连通图 G 和 m 种不同的颜色。用这些颜色为图 G 的各节点着色，对每个节点都着一种颜色。如果有一种着色方案可以使图 G 中每条边的两个节点着不同的颜色，则称这个图是 m 可着色的。计算图的不同的着色方案数。

输入：第 1 行包含 3 个正整数 n、k 和 m，表示有 n 个节点、k 条边和 m 种颜色。节点编号为 1～n。在接下来的 k 行中，每行都有两个正整数 u、v，表示在 u、v 之间有一条边。N≤100，k≤2500，保证答案不超过 20 000。

输出：单行输出不同的着色方案数。

输入样例	输出样例
5 8 4 1 2 1 3 1 4 2 3 2 4 2 5 3 4 4 5	48

题解：本题为**图的 m 着色问题**，可采用回溯法（m 叉树）解决。

描述（HDU2553）：在 N×N 的方格棋盘上放置 N 个皇后，使得它们不相互攻击（即任意两个皇后都不允许同行、同列，也不允许在与棋盘边框成 45 角的斜线上。求有多少种合法的放置方案。

输入：输入包含多个测试用例，每个测试用例都包含一个正整数 N（N≤10），表示棋盘和皇后的数量，如果 N=0，则表示结束。

输出：对每个测试用例，单行输出一个正整数，表示有多少种合法的放置方案。

输入样例	输出样例
1 8 5 0	1 92 10

题解：本题为 N 皇后问题，可采用回溯法（m 叉树或排列树）解决。