**[部分背包问题（贪心）](http://www.cnblogs.com/gzydn/archive/2009/05/21/1486102.html)**

所谓贪心算法是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，所做出的仅是在某种意义上的局部最优解。  
　　贪心算法不是对所有问题都能得到整体最优解，但对范围相当广泛的许多问题他能产生整体最优解或者是整体最优解的近似解。  
　　贪心算法的基本思路如下：  
　　1.建立数学模型来描述问题。  
　　2.把求解的问题分成若干个子问题。  
　　3.对每一子问题求解，得到子问题的局部最优解。  
　　4.把子问题的解局部最优解合成原来解问题的一个解。

**部分背包问题：**     给定一个最大载重量为M的卡车和N种食品，有食盐，白糖，大米等。已知第 i 种食品的最多拥有Wi 公斤，其商品价值为Vi元/公斤，编程确定一个装货方案，使得装入卡车中的所有物品总价值最大。

    分析：因为每一个物品都可以分割成单位块，单位块的利益越大显然总收益越大，所以它局部最优满足全局最优，可以用贪心法解答。方法如下：  
（1）先将单位块收益按从大到小进行排序；  
（2）初始化背包的剩余体积和当前价值；  
（3）从前到后考虑所有物品：a.如果可以完全放入，当前价值加上物品总价值，剩余体积减去物品总体积；b.如果可以部分放进，当前价值加上物品价值\*剩余体积，使剩余体积为0.

http://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifCode  
 1http://www.cnblogs.com/Images/OutliningIndicators/None.gifVar I   :Integer;  
 2http://www.cnblogs.com/Images/OutliningIndicators/None.gifBegin  
 3http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Assign(Input,Finp); Reset(Input);  
 4http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Readln(M,N);  
 5http://www.cnblogs.com/Images/OutliningIndicators/None.gif  For I:=1 To N Do Readln(W[I],P[I]);  
 6http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Close(Input);  
 7http://www.cnblogs.com/Images/OutliningIndicators/None.gifEnd;  
 8http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
 9http://www.cnblogs.com/Images/OutliningIndicators/None.gifProcedure Sort(L,R:Integer);   {用快速排序按收益值从大到小排序}  
10http://www.cnblogs.com/Images/OutliningIndicators/None.gifVar I,J,Y :Integer;  
11http://www.cnblogs.com/Images/OutliningIndicators/None.gif    X     :Real;  
12http://www.cnblogs.com/Images/OutliningIndicators/None.gifBegin  
13http://www.cnblogs.com/Images/OutliningIndicators/None.gif  I:=L; J:=R;  
14http://www.cnblogs.com/Images/OutliningIndicators/None.gif  X:=P[(L+R) Div 2]/W[(L+R) Div 2];  
15http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Repeat  
16http://www.cnblogs.com/Images/OutliningIndicators/None.gif    While (I<R)And(P[I]/W[I]>=X) Do Inc(I);  
17http://www.cnblogs.com/Images/OutliningIndicators/None.gif    While (P[J]/W[J]<=X)And(J>L) Do Dec(J);  
18http://www.cnblogs.com/Images/OutliningIndicators/None.gif    If I<=J Then  
19http://www.cnblogs.com/Images/OutliningIndicators/None.gif      Begin  
20http://www.cnblogs.com/Images/OutliningIndicators/None.gif        Y:=P[I]; P[I]:=P[J]; P[J]:=Y;  
21http://www.cnblogs.com/Images/OutliningIndicators/None.gif        Y:=W[I]; W[I]:=W[J]; W[J]:=Y;  
22http://www.cnblogs.com/Images/OutliningIndicators/None.gif        Inc(I); Dec(J);  
23http://www.cnblogs.com/Images/OutliningIndicators/None.gif      End;  
24http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Until I>J;  
25http://www.cnblogs.com/Images/OutliningIndicators/None.gif  If I<R Then Sort(I,R);  
26http://www.cnblogs.com/Images/OutliningIndicators/None.gif  If L<J Then Sort(L,J);  
27http://www.cnblogs.com/Images/OutliningIndicators/None.gifEnd;  
28http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
29http://www.cnblogs.com/Images/OutliningIndicators/None.gifProcedure Work;  
30http://www.cnblogs.com/Images/OutliningIndicators/None.gifVar I :Integer;  
31http://www.cnblogs.com/Images/OutliningIndicators/None.gifBegin  
32http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Sort(1,N);  
33http://www.cnblogs.com/Images/OutliningIndicators/None.gif  For I:=1 To N Do  
34http://www.cnblogs.com/Images/OutliningIndicators/None.gif    If M>=W[I] Then     {如果全部可取，则全取}  
35http://www.cnblogs.com/Images/OutliningIndicators/None.gif      Begin  
36http://www.cnblogs.com/Images/OutliningIndicators/None.gif        S:=S+P[I]; M:=M-W[I];  
37http://www.cnblogs.com/Images/OutliningIndicators/None.gif      End  
38http://www.cnblogs.com/Images/OutliningIndicators/None.gif    Else        {否则取一部分}  
39http://www.cnblogs.com/Images/OutliningIndicators/None.gif      Begin  
40http://www.cnblogs.com/Images/OutliningIndicators/None.gif        S:=S+M\*(P[I]/W[I]); Break;  
41http://www.cnblogs.com/Images/OutliningIndicators/None.gif      End;  
42http://www.cnblogs.com/Images/OutliningIndicators/None.gifEnd;  
43http://www.cnblogs.com/Images/OutliningIndicators/None.gifProcedure Out;    {输出}  
44http://www.cnblogs.com/Images/OutliningIndicators/None.gifBegin  
45http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Assign(Output,Fout); Rewrite(Output);  
46http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Writeln(S:0:0);  
47http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Close(Output);  
48http://www.cnblogs.com/Images/OutliningIndicators/None.gifEnd;  
49http://www.cnblogs.com/Images/OutliningIndicators/None.gifBegin      {主程序}  
50http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Init;  
51http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Work;  
52http://www.cnblogs.com/Images/OutliningIndicators/None.gif  Out;  
53http://www.cnblogs.com/Images/OutliningIndicators/None.gifEnd.

 利用贪心策略解题，需要解决两个问题：  
一、确定问题是否能用贪心策略求解；一般来说，适用于贪心策略求解的问题具有以下特点：  
① 可通过局部的贪心选择来达到问题的全局最优解。运用贪心策略解题，一般来说需要一步步的进行多次的贪心选择。在经过一次贪心选择之后，原问题将变成一个相似的，但规模更小的问题，而后的每一步都是当前看似最佳的选择，且每一个选择都仅做一次。  
② 原问题的最优解包含子问题的最优解，即问题具有最优子结构的性质。在背包问题中，第一次选择单位质量最大的货物，它是第一个子问题的最优解，第二次选择剩下的货物中单位重量价值最大的货物，同样是第二个子问题的最优解，依次类推。  
二、如何选择一个贪心标准？正确的贪心标准可以得到问题的最优解，在确定采用贪心策略解决问题时，不能随意的判断贪心标准是否正确，尤其不要被表面上看似正确的贪心标准所迷惑。在得出贪心标准之后应给予严格的数学证明。

**贪心一般都伴随着使用排序**