标签： [算法](http://www.csdn.net/tag/%e7%ae%97%e6%b3%95)[c](http://www.csdn.net/tag/c)

2012-07-06 17:09 184972人阅读 [评论](http://blog.csdn.net/mu399/article/details/7722810#comments)(61) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/mu399/article/details/7722810#report)

 分类：

01背包 动态规划

版权声明：本文为博主原创文章，未经博主允许不得转载。

01背包问题，是用来介绍动态规划[**算法**](http://lib.csdn.net/base/datastructure)最经典的例子，网上关于01背包问题的讲解也很多，我写这篇文章力争做到用最简单的方式，最少的公式把01背包问题讲解透彻。

**01背包的状态转换方程 f[i,j] = Max{ f[i-1,j-Wi]+Pi( j >= Wi ),  f[i-1,j] }**

f[i,j]表示在前i件物品中选择若干件放在承重为 j 的背包中，可以取得的最大价值。

Pi表示第i件物品的价值。

决策：为了背包中物品总价值最大化，第 i件物品应该放入背包中吗 ？

**题目描述：**

有编号分别为a,b,c,d,e的五件物品，它们的重量分别是2,2,6,5,4，它们的价值分别是6,3,5,4,6，现在给你个承重为10的背包，如何让背包里装入的物品具有最大的价值总和？

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | weight | value | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| a | 2 | 6 | 0 | 6 | 6 | 9 | 9 | 12 | 12 | 15 | 15 | 15 |
| b | 2 | 3 | 0 | 3 | 3 | 6 | 6 | 9 | 9 | 9 | 10 | 11 |
| c | 6 | 5 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 10 | 11 |
| d | 5 | 4 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 10 | 10 |
| e | 4 | 6 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

只要你能通过找规律手工填写出上面这张表就算理解了01背包的动态规划算法。

首先要明确这张表是至底向上，从左到右生成的。

为了叙述方便，用e2单元格表示e行2列的单元格，这个单元格的意义是用来表示只有物品e时，有个承重为2的背包，那么这个背包的最大价值是0，因为e物品的重量是4，背包装不了。

对于d2单元格，表示只有物品e，d时,承重为2的背包,所能装入的最大价值，仍然是0，因为物品e,d都不是这个背包能装的。

同理，c2=0，b2=3,a2=6。

对于承重为8的背包，a8=15,是怎么得出的呢？

根据01背包的状态转换方程，需要考察两个值，

一个是f[i-1,j],对于这个例子来说就是b8的值9，另一个是f[i-1,j-Wi]+Pi；

在这里，

 f[i-1,j]表示我有一个承重为8的背包，当只有物品b,c,d,e四件可选时，这个背包能装入的最大价值

f[i-1,j-Wi]表示我有一个承重为6的背包（等于当前背包承重减去物品a的重量），当只有物品b,c,d,e四件可选时，这个背包能装入的最大价值

f[i-1,j-Wi]就是指单元格b6,值为9，Pi指的是a物品的价值，即6

由于f[i-1,j-Wi]+Pi = 9 + 6 = 15 大于f[i-1,j] = 9，所以物品a应该放入承重为8的背包

以下是actionscript3 的代码

**[java]** [view plain](http://blog.csdn.net/mu399/article/details/7722810) [copy](http://blog.csdn.net/mu399/article/details/7722810)

1. **public** function get01PackageAnswer(bagItems:Array,bagSize:**int**):Array
2. {
3. var bagMatrix:Array=[];
4. var i:**int**;
5. var item:PackageItem;
6. **for**(i=0;i<bagItems.length;i++)
7. {
8. bagMatrix[i] = [0];
9. }
10. **for**(i=1;i<=bagSize;i++)
11. {
12. **for**(var j:**int**=0;j<bagItems.length;j++)
13. {
14. item = bagItems[j] as PackageItem;
15. **if**(item.weight > i)
16. {
17. //i背包转不下item
18. **if**(j==0)
19. {
20. bagMatrix[j][i] = 0;
21. }
22. **else**
23. {
24. bagMatrix[j][i]=bagMatrix[j-1][i];
25. }
26. }
27. **else**
28. {
29. //将item装入背包后的价值总和
30. var itemInBag:**int**;
31. **if**(j==0)
32. {
33. bagMatrix[j][i] = item.value;
34. **continue**;
35. }
36. **else**
37. {
38. itemInBag = bagMatrix[j-1][i-item.weight]+item.value;
39. }
40. bagMatrix[j][i] = (bagMatrix[j-1][i] > itemInBag ? bagMatrix[j-1][i] : itemInBag)
41. }
42. }
43. }
44. //find answer
45. var answers:Array=[];
46. var curSize:**int** = bagSize;
47. **for**(i=bagItems.length-1;i>=0;i--)
48. {
49. item = bagItems[i] as PackageItem;
50. **if**(curSize==0)
51. {
52. **break**;
53. }
54. **if**(i==0 && curSize > 0)
55. {
56. answers.push(item.name);
57. **break**;
58. }
59. **if**(bagMatrix[i][curSize]-bagMatrix[i-1][curSize-item.weight]==item.value)
60. {
61. answers.push(item.name);
62. curSize -= item.weight;
63. }
64. }
65. **return** answers;
66. }

PackageItem类

**[java]** [view plain](http://blog.csdn.net/mu399/article/details/7722810) [copy](http://blog.csdn.net/mu399/article/details/7722810)

1. **public** **class** PackageItem
2. {
3. **public** var name:String;
4. **public** var weight:**int**;
5. **public** var value:**int**;
6. **public** function PackageItem(name:String,weight:**int**,value:**int**)
7. {
8. **this**.name = name;
9. **this**.weight = weight;
10. **this**.value = value;
11. }
12. }

测试代码

**[java]** [view plain](http://blog.csdn.net/mu399/article/details/7722810) [copy](http://blog.csdn.net/mu399/article/details/7722810)

1. var nameArr:Array=['a','b','c','d','e'];
2. var weightArr:Array=[2,2,6,5,4];
3. var valueArr:Array=[6,3,5,4,6];
4. var bagItems:Array=[];
5. **for**(var i:**int**=0;i<nameArr.length;i++)
6. {
7. var bagItem:PackageItem = **new** PackageItem(nameArr[i],weightArr[i],valueArr[i]);
8. bagItems[i]=bagItem;
9. }
10. var arr:Array = ac.get01PackageAnswer(bagItems,10);