背包问题(动态规划—自底向上)

部分程序：

public static int[,] result=new int[11,4];

public static int BottomUp(int m, int i, int[] w, int[] p)

{

if (result[m, i] != 0) return result[m, i];

for (int tempM = 1; tempM < m+1; tempM++) /\*循环从1到m,一共是m次，m是重量\*/

{

for (int tempI = 1; tempI < i+1; tempI++) /\*从1到i，一共是i次\*/

{

if(result[tempM,tempI]!=0)continue;

if (w[tempI] > tempM)

{

result[tempM, tempI] = result[tempM, tempI - 1];

}

else

{

int maxValue1 = result[tempM - w[tempI], tempI - 1] + p[tempI];

int maxValue2 = result[tempM, tempI - 1];

if (maxValue1 > maxValue2)

{

result[tempM, tempI] = maxValue1;

}

else

{

result[tempM, tempI] = maxValue2;

}

}

}

}

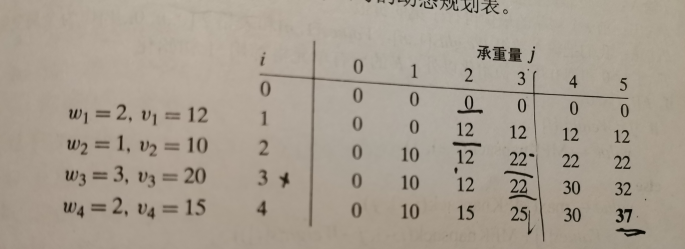
return result[m, i];

}

}

}

时间复杂度为O(i\*m),把i换成n，O(n\*m),空间复杂度也是O(n\*m)



对于一张填好的动态规划表，只需在对应最大重量的列寻找最优子集，用一个循环即可，所以时间属于O(n)