背包问题

写在前面：作为最经典的算法题目之一，背包问题本来就是多种多样的，比如01背包问题，完全背包问题，多重背包问题。对于不同的背包问题，不同的数据范围的背包问题有有不同的解法，当然，对于数据范围大的同类的背包问题的解法适用数据范围小的背包问题。

当然我的背包问题系列问题是跟着网络上的dd大神的背包九讲学来的，可惜说其中的内容让我对背包问题的各种变体有了深刻的理解。

今天的实验内容当然离不开搜索，所以我选择数据范围非常小的一道01背包问题，它的就是用最简单的搜索就能解决。

下面就直接贴我的代码：

#include<iostream>

using namespace std;

const int N = 100;

int n, w;

int W[N], V[N];

int max(int a, int b)

{

return a > b ? a : b;

}

//从第i个物品开始挑选总重小于j的部分

int solve(int i, int j)

{

int res;

if (i == n)//已经没有剩余的物品了

res = 0;

else if (j < W[i])

res = solve(i + 1, j);

else

{

res = max(solve(i + 1, j), solve(i + 1, j - W[i]) + V[i]);//一个物品选还是不选都试一下

}

return res;

}

int main()

{

cin >> n >> w;

for (int i = 0; i < n; i++)

cin >> W[i];

for (int i = 0; i < n; i++)

cin >> V[i];

cout << solve(0, w) << endl;

return 0;

}