设计：

最初始状态（都没有使用优惠组合）：

C（A1，A2,A3,A4,A5）

使用优惠组合（动态规划）：

用Bi表示第i种商品使用优惠组合

C（A-B1，A2-B2,A3-B3,A4-B4,A5-B5）态比初始状态优惠存在最优子结构优惠组合

可以多次使用优惠组合在子问题

分析：

按照此优惠组合递归式，用动态规划算法设计出来。其中product[i];是购买的第i种商品。

private static void minicost(){

int i,j,k,m,n,p,minm;

minm = 0;

for(i=1; i<=b; i++)

minm += product[i]\*purch[i].price;

for(p=1; p<=s; p++){

i = product[1] - offer[p][1];

j = product[2] - offer[p][2];

k = product[3] - offer[p][3];

m = product[4] - offer[p][4];

n = product[5] - offer[p][5];

if(i>=0 && j>=0 && k>=0 && m>=0 && n>=0 &&cost[i][j][k][m][n]+offer[p][0] < minm)

minm = cost[i][j][k][m][n] + offer[p][0];

} cost[product[1]][product[2]][product[3]][product[4]][product[5]] = minm;

}