这道题还是要转化一下为最大上升子序列问题，01背包性质的解法

先对物品从小到大**排序**，为了方便之后的计算

创建二维数组dp[2010][2010]

其中第一维表示的是当前物品种类，第二维表示当前对数

所以状态转移方程为

dp[i][j]=min(dp[i-1][j],dp[i-2][j-1]+(th[i]-th[i-1])\*(th[i]-th[i-1]));

如dp[3][1]= min(dp[2][1],dp[1][0]+(th[3]-th[2])\*(th[3]-th[2]));

表示求当前为第三种物体，目前有一对时候的状态转移

1. 保持原来不变，之前两种物品时候已经存在一对了
2. 第一种1种0对时的状态加上2,3种物品的疲劳度

这样问题的性质就非常像0-1背包问题的求法

总结：目前遇到的很对dp问题如猴子堆塔问题，都需要对数组进行排序预处理，然后解法有点偏向于LIS的解法，简化了问题。