1、

（1） 尽可能充满背包，顺序为A、B、C，总价为8，总重15。

（2）挑贵的，顺序为E、B、C、D或E、C、B、D，总价为15，总重为8。

（3）挑价重比高的，顺序为E、C、B、D或E、C、D、B，总价为15，总重为8。

注意：要写清楚顺序。

（4）满足约束：，求解目标函数：的情况。

注意：目标函数一定要取最大值，否则与题意不符。

（5）计算每件物体的价重比，并降序排列各物品后执行：

int sum=0, value=0;

for(int i=1;i<=N;i++)

{

if(sum+<=)

{

sum+=;

=1;

value+=

}

}

print(“Whole value: %d”, value );//总价即为

注意：意思相近即可。但，可能会有超重物体在取的过程中被略去；也可能最后背包有剩余空间但放不下任何物品；答案要与（4）所描述的题目相符。

2、O(n2) ；500

3、

(1)、伪代码如下：

for i=1 to N-1:

haschanged=false

for j=1 to N-i：

if A[j]>A[j+1]:

swap(A[j], A[j+1])

haschanged=true

if not haschanged：

break

得分要点：

1.外层循环次数N-1

2.内层循环次数与外层的循环的轮次相关，轮次为i时，循环次数为N-i

(2)、O(n2)

注意：必须使用渐进记号表示，可不写出推导过程。

4、

算法思想：

将大数据集划分为多个可被内存处理的子集合，分别进行排序。后进行归并，在内存中同时将数个子集合的部分归并，排序为一个更大的已排序子集合。依次归并直到最终将所有数据归并为一个已排序集合。

基本过程：

1、将数据划分为数个可被内存处理的子集合，装入内存进行内排序。

2、将数个已排序子集合的一块磁盘块装入内存块并留出排序区。

3、将每个子集合当前第一个元素装入待比较块对应位置，选择比较区中最小者装入待输出块，并从该元素原子集合提取下一个元素进入待比较块。每当待输出块装满则写入外存。若内存中一块被处理完，则从外存中写入下一块。

4、若一次归并后还未处理完数据，则将经过归并的子集合作为新的子集合继续归并，直到最终完成。

注意：按照题目要求回答，必须是外部排序中的归并。