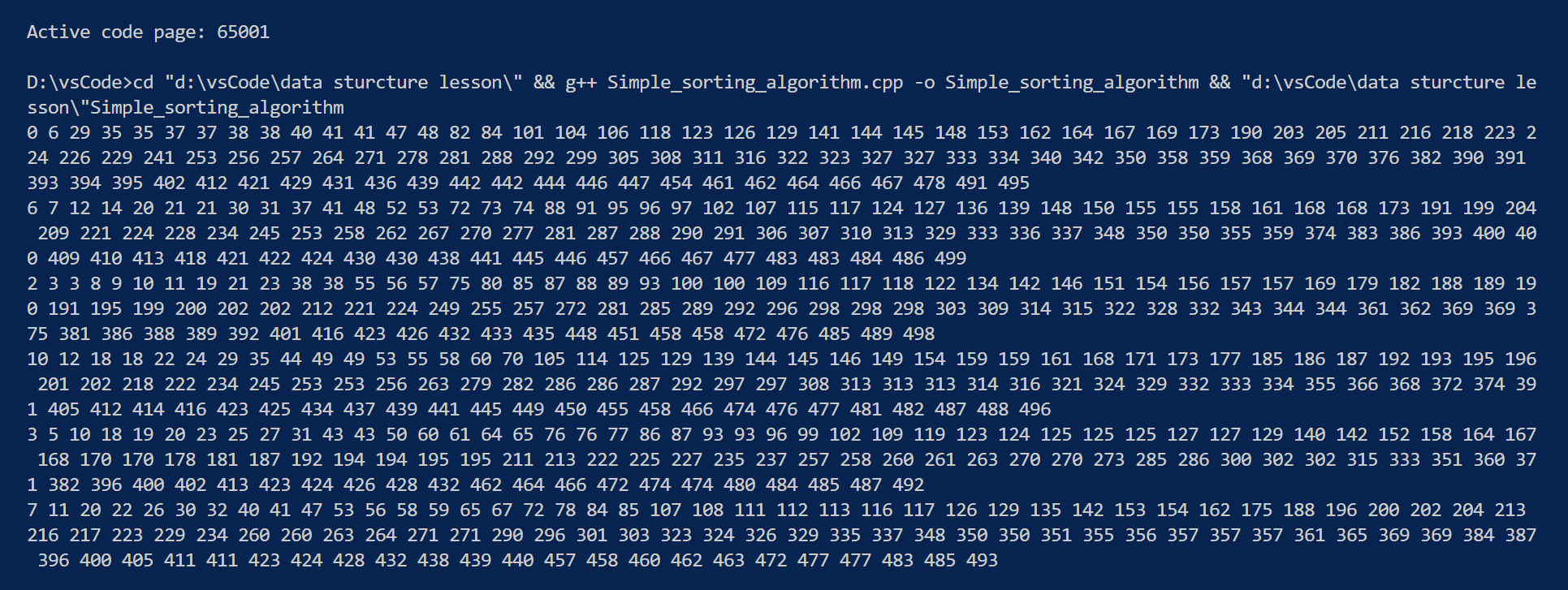
一．数据结构与算法的关系：算法是建立在数据结构上的，只有建立了相应的结构才能写好算法，可以说数据结构是算法的基石；而若没有算法，单独只有一个数据结构并无意义，因此，数据结构与算法缺一不可。

二．见Simple\_sorting\_algorithm.cpp

测试数据：通过随机数生成

测试结果：

（直接）冒泡排序算法：

时间复杂度：O(n²)，因为冒泡排序嵌套了2层循环，外层循环执行了n-1次，第i次循环执行耗时O(n-i-1)，累加后可得其时间复杂度即为O(n\*n)。

空间复杂度：O(1) , 为一个[常量](https://baike.baidu.com/item/%E5%B8%B8%E9%87%8F?fromModule=lemma_inlink" \t "_blank)，不随被处理数据量n的大小而改变。

（直接）选择排序算法：

时间复杂度：O(n²)，同理选择排序也嵌套了2层循环，外层循环执行了n-1次，第i次循环执行耗时O(n-i-1)，累加后可得其时间复杂度也为O(n\*n)。

空间复杂度：O(1)，为一个[常量](https://baike.baidu.com/item/%E5%B8%B8%E9%87%8F?fromModule=lemma_inlink" \t "_blank)，不随被处理数据量n的大小而改变。

（直接）插入排序算法：

时间复杂度：O(n²)，循环共执行n-1次，第i次循环时最多要进行i-1次比较，最多共需进行次比较，因此其时间复杂度为O(n\*n)。

空间复杂度：O(1)，为一个[常量](https://baike.baidu.com/item/%E5%B8%B8%E9%87%8F?fromModule=lemma_inlink" \t "_blank)，不随被处理数据量n的大小而改变。

（直接）冒泡排序的递归算法：

时间复杂度：O(n²)，因为递归深度为n-1，每次要进行一个n-i-1次的循环，相乘可得其时间复杂度即为O(n\*n)。

空间复杂度：O(n)，递归深度为n-1，每次递归调用规模为1,空间复杂度即为 O(n)。

（直接）选择排序的递归算法：

时间复杂度：O(n²)，同理因为递归深度为n-1，每次要进行一个n-i-1次的比较，相乘可得其时间复杂度为O(n\*n)。

空间复杂度：O(n)，递归深度为n-1，每次递归调用规模为1,空间复杂度即为O(n)。

（直接）插入排序的递归算法：

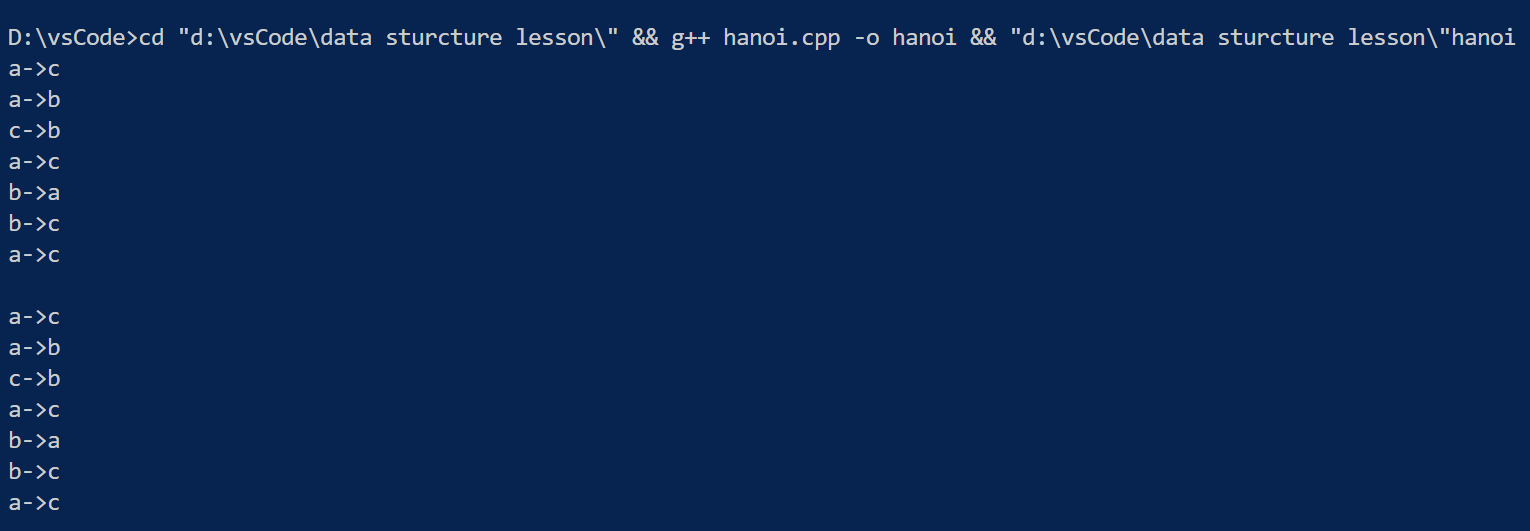
时间复杂度：O(n²)，同理因为递归深度为n-1，每次递归最多需进行次比较，相乘可得其时间复杂度为O(n\*n)。

空间复杂度：O(n)，递归深度为n-1，每次递归调用规模为1,空间复杂度即为 O(n)。

1. 见hanoi.cpp

测试数据：n=3

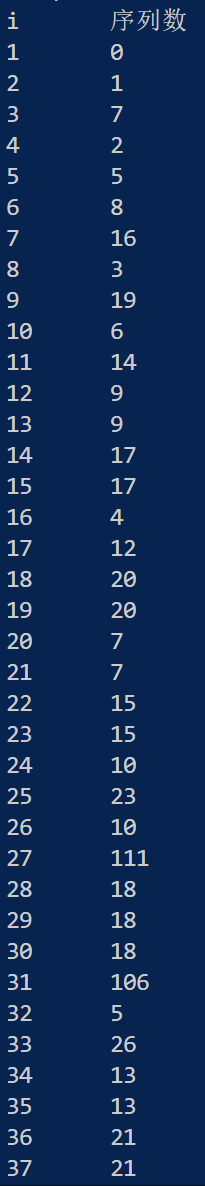
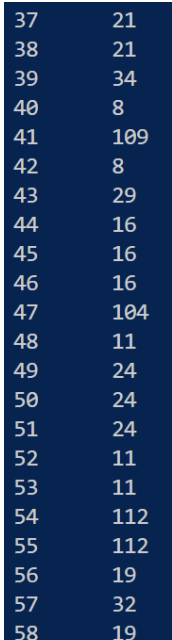
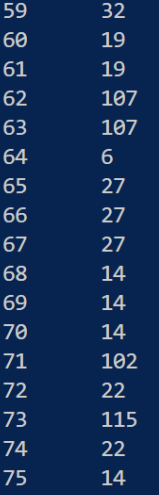
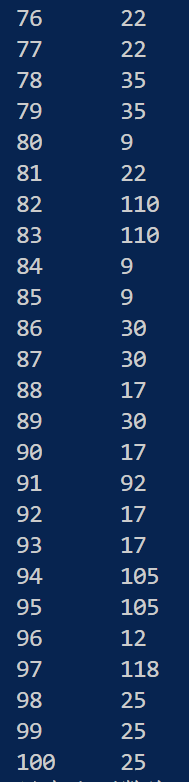
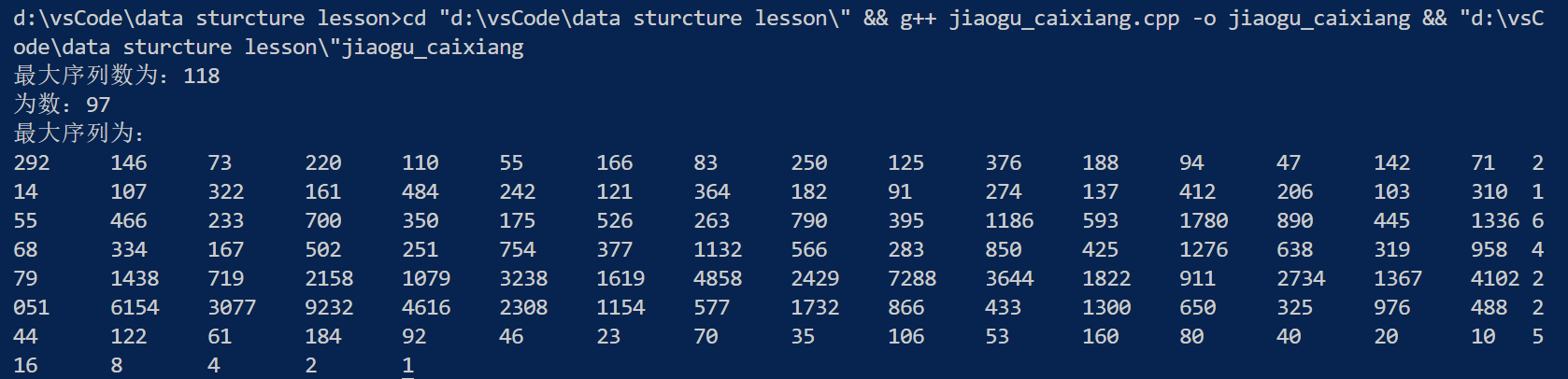
测试结果：



时间复杂度：若问题规模为n，汉诺塔的递归算法则需进行两次n-1规模问题的操作，即T(n)=2T(n-1)，同理进行迭代，得到T(n)=2^n\*T(1)，其中T(1)为一个常数，故其时间复杂度为O(2^n)。

空间复杂度：O(2^n)。

非递归用栈的形式模拟，本质上是一样的。所以时间和空间复杂度相同。

1. 1.测试结果：见jiaogu\_caixiang.cpp
2. 角谷猜想是正确的。