第三次作业

2018.9.25

3-1

设：

是给定序列，是序列中的第个元素；

为以结尾的单调递增子序列的长度；

为子序列中的前一个元素的下标,若是子序列的第一个，那么.

则有:

通过计算确定。

首先计算所有的；

可以通过一次遍历得到最长的子序列长度,并通过*B*推出整个子序列。

3-2

设：

输入的序列为数组A；

定义数组B，其中B[i]代表长度为（i+1）的子序列的最小结尾元素。

算法思路如下：

①B[0]初始化为A[0];

②从A[1]开始，对A的每一个元素A[i]依次和B[n]比较：

a.若B[n]小于A[i],说明子序列长度可以增加，B[n+1]=A[i];

b.否则,说明子序列长度不能增加，但需要更新B；从B[0]到B[n],找到第一个比A[i]大的数B[j]，用A[i]替换B[j];

在构造过程中，数组B是有序的，所以搜索A[i]位置可以用二分查找来实现，复杂度O(logN).

所以整个算法的复杂度是O(NlogN).

B的有定义的元素个数即为最长递增子序列长度。

3-5

设：

m(i,j,k)为可选择物品为第1，2，…，i个;

总容量为j;

总容积为k的背包能装的最大价值;

结合书本，有递归式：

算法复杂度为O(nCD),n为物品个数，C为容量，D为容积。