LIS_T说明

给定一个排列，1-n中任何一个数对应于树上一个节点，根为0元素，***根到节点i的有序路径表示以i这个值为结尾的字典序最小的最长LIS方案***，其构造方法如下：

维护一个**单调迭代数组**(是这样一个数据结构，不妨设单调递增，支持末端插入元素的动态数组，单点替换，修改操作不影响数组单调性)，s[1..cnt]，其中某个时刻T时，插入元素x，**lower_bound**查找s中第一个严格大于x的下标i，s[i]由y替换为x(y>x)表示以值x结尾的LIS长度为i，且存在一个***字典序最小***的方案，方案中x的前驱元素应该是此时的s[i-1]，利用前驱关系构建出来的显然是一棵有序树(**儿子们相对有序**)，***元素y被x替代，表明x和y在同一层，且x在y的右侧***；这个数组在某个时刻的数学意义是这棵树的右侧投影节点编号，这棵树的深度是整个排列的LIS长度，同层元素递减；

这个结构可以很方便的证明排列的***最小递增子序列的切割定理***，一个排列最少可以分成k个单调递减的子序列，其中k为最长上升序列的长度；这是因为，*只要把树上每层的元素拿出来，就是一个方案*，这表明k至多是LIS长度，而树上深度最大的叶子节点，显然要至少深度个递减序列覆盖，所以k至少是LIS长度，故k=len(LIS)得证；