

第 9 次大作业

算法思路如下：在内存中申请一定的内存，保存若干个栈，初始时设定栈为空，栈数量 pos 为 0。顺序读取输入序列，如果输出栈数量为 0 或输入比输出栈数组末端栈顶元素大则栈数量加一，输入压入输出栈数组的末端栈；否则对输出栈顶元素数组进行二分查找，找到第一个栈顶元素比输入大的栈，将输入压入之。如此，通过降低输出栈顶元素数组的整体大小保证可以有最多的元素加入该子序列，即可求出 LIS 长度。输出子序列时回溯，反向遍历输入数组，同时反向遍历栈数组，在每个栈中查找最后一个入栈且尚未被跳过的元素输出。

复杂度分析：遍历+二分查找的时间复杂度为 $O(n\log n)$ 。由于事先不知道输出的规模，因此需要 $O(n^2)$ 的临时空间，空间复杂度为 $O(n^2)$ 。