排序

- 基于插入
    - 直接插入：未排序的数依次插入已排序数组，逐步后移，O(n^2)
    - 希尔排序：时间复杂度：O(n^1.3)，空间复杂度O(1)
    - 选择排序
        - 直接排序，先找最小，再第二小，依次类推
        - 堆排序：O(nlogn)，空间O(n)
- 基于交换
    - 冒泡排序，O(n^2),空间O(1),左右比较将大的后移，最大的放在最后，之后继续
    - 快速排序，O(nlogn), 空间O(nlogn)，选取基准，大的放右边小的左边，二分思想
- 归并排序，O(nlogn)，空间0（n），二叉树分而治之
- 基数排序：百十个位比较

数据结构

- 集合结构
- 线性结构
    - 顺序表：
        - 查找元素O(1),插入删除O(N)
    - 链表(节点+数据)
        - 查找元素O(N),插入删除O(1)
    - 栈: 后入先出LIFO，栈顶指针
    - 队列：先入先出FIFO，头指针+尾指针
        - 环形队列： 队空： front=rear，队满：(rear+1)%maxsize=front
- 树结构
    - 父亲节点->孩子节点，没有孩子的节点是叶子节点，从根节点到叶子节点的最长路径称为度（或者深度）
    - 二叉树： 每个非叶节点至多只有两个孩子的数
        - 深度：logN-N
        - 满二叉树：某一节点的编号是N，则两个孩子节点的序号分别是2N和2N+1 ->二叉树线性存储，总共需2^N空间
    - 遍历：前序：中左右；中序；后序
        - 中序+任一序->另一序

    - 平衡二叉树AVL：每个节点左子树的度和右子树的度相差不超过1
    - 堆：一种完全二叉树
        - 大/小顶堆：每个结点的值都大于或等于其左右孩子结点的值
- 图结构
    - 有向图 && 无向图
        - 强联通图：有向图中任意两顶点可以相互到达，弱联通图
    - 有环图 && 无环图
    - 存储方式
        - 集合方式：顶点集合+边集合
        - 矩阵方式：一个矩阵存储每两个节点之间的代价
    - 遍历方式： DFS：栈，依次弹出顶点遍历；BFS：队列