4/21

二叉排序树，也称二叉搜索树，特点是 左子树的节点值 < 根节点值 < 右子树的节点值。所以最小值的左子树为空。

堆排序平均执行的时间复杂度和需要附加的存储空间复杂度分别是 O(nlog(n))和O(1)

快排平均：T(n)=2T(n/2)+n ==>>T(n)=nlog(n)

堆排序平均：每次下滤，两个儿子比较，然后与父亲比较，因为一般下滤logn层，每次共比较2logn，所以n\*2logn

4/22

对AOV网进行拓扑排序的方法如下：

(1)在AOV网中选择一个入度为零(没有前驱)的顶点且输出它；

(2)从网中删除该顶点及与该顶点有关的所有边；

(3)重复上述两步，直至网中不存在入度为零的顶点为止。

在一棵二叉排序树上查找值为35的数据，比较的数据序列正确的为

46、36、18、28、35

在含有10个结点的二叉排序树上，查找关键字为20的结点，则依次比较的关键字有可能是

A：首先根节点为25,20比25小搜索其左子树，找到10比25小不矛盾，20比10大搜索其右子树，找到15比10大不矛盾，20比15大搜索其右子树找到20，正确

B：首先根节点25,20比25小搜索其左子树，找到10比25小不矛盾，20比10大搜索其右子树，找到15比10大不矛盾，

20比15大搜索其右子树，找到18比15大不矛盾， 20比\*\*\*搜索其右子树，找到20，正确

C：首先根节点为10,20比10大搜索其右子树，找到30比10大不矛盾，20比30小搜索其左子树，找到20，正确

D：首先根节点为10, 20比10大搜索其右子树，找到30比10大不矛盾，20比30小搜索其左子树，找到25比30小不矛盾，20比25小搜索其左子树找到20，正确

二叉排序树（Binary Sort Tree）又称二叉查找树.它或者是一棵空树；或者是具有下列性质的二叉树：（1）若左子树不空,则左子树上所有结点的值均小于它的根结点的值； （2）若右子树不空,则右子树上所有结点的值均大于它的根结点的值； （3）左、右子树也分别为二叉排序树

错误， 5

/ \

3 6

\ /

8 2

这棵二叉树满足其任一节点的值均大于其左孩子的值，小于右孩子的值，但它并不是二叉排序树,因为在右支路中有2<5,在左支路中8>5不满足二叉排序树的性质。

4/23

假设你只有100MB的内存，需要对1GB的数据进行排序，最合适的算法是

多路归并排序 当内存不足以一次性存入全部数据时，应使用外部排序

直接选择排序中存在着不相邻元素之间的互换，因此，直接选择排序是一种不稳定的排序方法。 A错

B:哈夫曼树定义

给定n个权值作为n个叶子结点，构造一棵二叉树，若带权路径长度达到最小，称这样的二叉树为最优二叉树，也称为哈夫曼树(Huffman

Tree)。哈夫曼树是带权路径长度最短的树，权值较大的结点离根较近。B对

C:我查找资料说有序指的是

不是结点的值有序,是结点的逻辑先后关系保持有序 C错

D:属于堆排序过程

计数排序是一种基于 统计 的排序算法

计数排序：取到待排序的长度为n的序列中的最大最小数（不妨假设都是非负数，若有负数，简单操作一下就可得到都是正数），这花费n的时间；

假设最大数为k，开辟一个数组大小为k的数组a[k]，若序列中有一个树为i，则把a[i]赋予标记，花费k的时间；

最后按顺序输出有标记的下标就可完成排序。

原地排序是指不申请多余空间排序，松一点的说法是可以用很小的固定的辅助空间

序列{2,1,4,9,8,10,6,20}是某排序算法第二轮排序的结果，则该算法只能是

快速排序

对n个元素用插入法建堆的时间复杂度是O(nlog(n))

快速排序在最坏情况下的时间复杂度与下面（ ）算法最坏情况下的时间复杂度相同

Shell 排序 冒泡排序

一趟排序结束后不一定能够选出一个元素放在其最终位置上的是 希尔排序

下列哪些算法在排序过程中需要一个记录的辅助空间

直接选择排序 直接插入排序 冒泡排序

前序 中序 后序 这三个两辆组合 必须要有 中序 才能唯一确定一棵二叉树 前与后 不能唯一确定

外部排序指待排序文件较大，内存一次性放不下，需存放在外部介质中。外部排序通常采用归并排序法

若外部存储上有3110400个记录，做6路平衡归并排序，计算机内存工作区能容纳400个记录，则排序好所有记录，需要作几趟归并排序

每次将工作区装满，共计3110400/400=7776个归并段，对于n路归并排序，m个归并段，需要归并排序的次数为log(n,m)次，代入数据得到答案为5

4/24/2023

经验证明，在所有同数量级的此类（先进的）排序方法中，快速排序的常数因子k最小。因此，就平均时间而言，快速排序是目前被认为是最好的一种内部排序方法

快速排序是在比较排序中平均性能最好的，但还有线性时间排序啊，比如：基数排序，计数排序，桶排序

0~999999之间的所有数字中，任何一位都不包括数字3的数字的总数为

9+8\*9+8\*9\*9+8\*9\*9\*9+8\*9\*9\*9\*9+8\*9\*9\*9\*9\*9=531441种

每位上可以取不是3的所有数，即可以取0,1,2,4,5,6,7,8,9，共九种情况。所以总数为：9\*9\*9\*9\*9\*9 = 9^6

= 531441种

对n个数字进行排序，期中两两不同的数字的个数为k，n远远大于k，而n的取值区间长度超过了内存的大小，时间复杂度最小可以是

两两不同的数字的个数为k，因为不知道确定的数字范围，桶排序不适合，又因为

n远远大于k，

本题使用hash表来统计，首先获得k个数及其每个数出现的次数，然后对k个数进行排序，排序的 复杂度可忽略不计，实际上就是遍历一遍n个数字，所以本位复杂度为O（n）

排序时，若不采用计数排序的等空间换时间的方法，则合并m个长度为n的已排序数组的最好时间复杂度为

O(mn(logm))

当n=1时，就成了m个数的归并排序，时间复杂度为O(mlogm)

设有 1000 个基本有序的元素，希望用最快的速度挑选出其中前 10 个最大的元素，最好选用（ ）排序法

冒泡排序

1、因为是topN的问题，所以一般考虑“选择排序”算法，这里只有“冒泡”是选择排序。

2、虽然冒泡的时间复杂度是O(n^2),但在这里由于基本有序，且只挑选前10个元素，复杂度10n左右，而b、c、d都是要做全局的排序，没有利用“基本有序”这个特点。

3、如果没有基本有序的条件，取topN，改进的“快排”，平均效果会更好些。”

因为只要前十个最大的，而且数据基本有序，所以此时不必要全部排序，用冒泡排序10趟就能将最大的10个上浮到最后面。

整数0x12345678，在采用bigendian中内存的排序序列是

12 34 56 78

小端模式：（从低字节到高字节）

地位地址 0x78 0x56 0x34 0x12 高位地址

大端模式：（从高字节到低字节）

地位地址 0x12 0x34 0x56 0x78 高位地址

一般默认大端所以大端原样小端交换前后

对关键码序列28,16,32,12,60,2,5,72快速排序（最常用的快速排序，以第一个关键码为基准）,使用挖坑法，从小到大一次划分结果为

(5,16,2,12)28(60,32,72)

京东商城plus会员的消费记录金额分别为900，512，613，700，810，若采用选择排序算法对其进行从小到大的排序，第三趟排序结果为

512，613，700，900，810