

原 数据结构知识点汇总

- 1、用链表表示线性表的优点是（便于插入和删除操作）
- 2、单链表中，增加头结点的目的是（方便运算的实现）
- 3、栈和队列的共同特点是（只允许在端点处插入和删除元素）
- 4、栈通常采用的两种存储结构是（线性存储结构和链表存储结构）
- 5、队列具有（先进先出）的特征，栈具有（后进先出）的特征。
- 6、链表（插入和删除不需要移动元素，但是无法随机访问任一元素）
- 7、循环链表的主要优点是（从表中任一结点出发都能访问到整个链表）
- 8、线性表（除了第一个和最后一个元素外，其余每个元素都有一个直接前驱和直接后继）
- 9、线性表的顺序存储结构和线性表的链式存储结构分别是（随机存取的存储结构、顺序存取的存储结构）

- 10、深度为5的满二叉树中，叶子结点的个数为（16）。 2^{k-1} 其共有（31）个结点。 $2^n - 1$

设一棵完全二叉树共有699个结点。则该二叉树的叶子结点数为（350）个。

#完全二叉树总的结点数为N，若N为奇数，则叶子结点数为 $(N+1)/2$ ；若N为偶数，则叶子结点数为 $N/2$ 。

- 11、具有3个结点的二叉树有（5）种形态。#高度为2层的是：根-左-右。高度为3层的是：根-左-左、根-左-右、根-右-右、根-右-左。

- 12、一棵二叉树中有3个叶子结点，有8个度为1的结点，则该二叉树中总的结点数为（13）个。

#叶子结点数 n_0 与度为2的结点数 n_2 的关系是： $n_0 = n_2 + 1$ ，所以度为2的结点个数为 $3-1=2$ 。所以总的结点数为 $n = n_0 + n_1 + n_2$, $8+2+3=13$ 。

- 13、已知二叉树的后序遍历序列是dabec,中序遍历序列是debac,它的前序遍历序列是（cedba）。#过程见文章：[点击打开链接](#)

- 14、已知二叉树的前序遍历序列是abdgcefh,中序遍历序列是dgbacfh,它的前序遍历序列是（gdbefhca）。

- 15、算法是指（解决方案的准确而完整的描述）。

- 16、算法由（顺序、选择、循环）控制结构组合而成。

- 17、算法的时间复杂度是指（算法执行过程中所需要的基本运算次数）。

- 18、算法的空间复杂度是指（执行过程中所需要的存储空间）。

- 19、算法分析的目的是（分析算法的效率以求改进）。

- 20、数据的存储结构是指（数据的逻辑结构在计算机中的表示）。

- 21、数据的逻辑结构是指（反映数据元素之间逻辑关系的数据结构）。

- 22、根据数据结构中各元素之间前后件关系的复杂程度，可将数据结构分为（线性结构和非线性结构）。

线性结构一般是首无前驱，尾无后继，中间元素有唯一的前驱和后继。主要有：列表、链表、队列、栈。

非线性结构主要有1、没有对应关系的集合。2、一对多关系的树。3、多对多关系的图。

- 23、（队列，循环队列，顺序表）不具有记忆功能，（栈）具有记忆功能。

- 24、递归算法一般需要用（栈）来实现。

#在递归算法的运行过程中，需要利用栈来保存其运算结果、参数和返回地址等。

- 25、算法的五个基本特征是：可行性，确定性，和拥有足够的情报

有限性：算法在执行有限步后必须终止。

确定性：算法的每个步骤都需要精确地定义，严格地、无歧义的运行。

输入：算法在运行之前赋给它的量。

输出：算法运行结束时的结果。

可行性：算法原则上能够精准地运行，而且人们用纸和笔做有限次运算后即可完成。

26、由两个栈共享一个存储空间的好处是（节省存储空间，降低上溢发生的概率）。

为了不发生上溢错误，就必须给每个栈分配一个足够大的存储空间。但实际中，很难准确地估计，若每个栈都分配过大的存储空间，势必造成系统空间个栈共用一个足够大的连续存储空间，则可利用栈的动态特性使它们的存储空间互补

27、需要打印机输出数据时，一般将打印作业放在一个（队列）中。

28、非空的循环单链表head的尾结点（由p所指向），满足（ $p \rightarrow next = head$ ）。

29、与单链表相比，双向链表的优点是（更容易访问相邻结点）。



30、

31、N个顶点的连通图中边的条数至少为（N-1）条。#将所有顶点连成一条线即可

32、N个顶点的强连通图中边的条数至少为（N）条。#将所有顶点连成一条圈

33、对长度为n的线性表进行顺序查找，最坏情况下需要比较（N）次。

34、最简单的交换排序是（冒泡排序）。

35、对长度为n的线性表进行顺序冒泡排序，最坏情况下需要比较（ $n(n-1)/2$ ）次。

#一共比较n-1遍，第1遍需要比较n-1次，第1遍需要比较n-2次，.....最后一遍需要比较1次。是一个等差序列，对其进行求和即可。

36、在序列基本有序的情况下，效率最高的方法是（A）#如果将插入排序换为冒泡排序，则选冒泡排序

A．插入排序 B．选择排序 C．快速排序 D．堆排序

插入排序通过数据元素的交换来逐步消除线性表中的逆序，所以比较的次数与初始排列次序有关，在待排序的元素序列基本有序的前提下，效率最高；选择排序和堆排序的比较次数与初始排列次序无关。快速排序虽然与初始排列次序有关，但在待排序的元素序列基本有序的前提下，效率低于插入排序。

37、希尔排序属于（插入类排序），堆排序属于（选择类排序）。

38、在下列几种排序方法中，要求内存量最大的是（D）。

A．插入排序 B．选择排序 C．快速排序 D．归并排序

快速排序的基本思想是，通过一趟排序将待排序记录分割成独立的两部分，其中一部分记录的关键字均比另一部分记录的关键字小，再分别对这两部分进行排序，以达到整个序列有序；插入排序的基本操作是指将无序序列中的各元素依次插入到已经有序的线性表中，从而得到一个新的序列；选择排序是：扫描整个线性表，从中选出最小的元素，将它交换到表的最前面(这是它应有的位置)，然后对剩下的子表采用同样的方法，直到表空为止；归并排序是将两个以上的有序表组合成一个新的序列表。

39、已知数据表A中每个元素距其最终位置不远，为节省时间，应采用（直接插入排序）。

40、数据结构是指相互有关联的(数据元素)的集合。

41、数据元素之间的任何关系都可以用(前驱和后继)关系来描述。

42、顺序存储方法是把逻辑上相邻的结点存储在(物理位置)相邻的存储单元中。

43、栈的基本运算有三种：入栈、退栈与读栈顶元素。

44、队列主要有两种基本运算：入队和退队。

45、在实际应用中，带链的栈可以用来收集计算机存储空间中所有空闲的存储结点，这种带链的栈称为(可利用栈)。

46、栈和队列通常采用的存储结构分别是 链式存储和顺序存储。

47、当循环队列非空且队尾指针等于对头指针时，说明循环队列已满，不能进行入队运算。这种情况称为（上溢）

48、当循环队列为空时，不能进行退队运算，这种情况称为（下溢）。

49、在一个容量为 25 的循环队列中，若头指针 $front=16$ ，尾指针 $rear=9$ ，则该循环队列中共有 18 个元素。注：当 $rear < front$ 时，元素 - $(front - rear)$ ；当 $rear > front$ 时，元素个数 = $rear - front$ 。

50、判断一个链表是否存在环：[点击打开链接](#)

单链表中元素的反转：[点击打开链接](#)

判断两个数组中是否有相同的数字：[点击打开链接](#)

从一个子序列中找出其最大子序列的和：[点击打开链接](#)

按单词反转字符串：[点击打开链接](#)

删除数组中重复的元素：[点击打开链接](#)

1、数组和链表的区别

数组不允许动态地定义其大小，只能将其定义成足够大小，这样可能会造成空间的浪费。

数组在内存中是顺序的存储，可以以 $O(1)$ 时间查找元素，但是需要 $O(n)$ 时间插入和删除元素（因为其后面的元素都需要跟着移动）。

链表可以动态地定义其大小。其在内存中是链式的存储，访问元素是需要从开头开始向后顺序访问，所以需要 $O(n)$ 时间查找元素；如果在所需位置直元素，需要 $O(1)$ 时间，如果在需要先找到所需位置再插入或删除元素，需要 $O(n)$ 时间。

2、链表的基本操作：反转，是否存在环，循环链表[点击打开链接](#)和双向链表[点击打开链接](#)的查找、插入、删除操作。

3、栈的入栈和出栈：[点击打开链接](#)，队列的入队和出队：[点击打开链接](#)

4、二叉树的基础知识：[点击打开链接](#)及其三种遍历（递归和非递归实现）：[点击打开链接](#)

5、图的基础知识：[点击打开链接](#)

6、常用散列函数和冲突消解机制：[点击打开链接](#)

7、排序算法中基本的冒泡排序、选择排序、插入排序需要很快地用代码实现。堆排序、归并排序、快速排序需要掌握其主要思想，并熟悉常用排序算法复杂度，及其应用范围：

(1) 当数据规模较小时，直接采用直接插入排序或直接选择排序。

(2) 当数据已经基本有序时，可以用直接插入排序或冒泡排序。

(3) 当数据规模较大时，可以用快速排序。当记录随机分布时，快速排序的平均时间最短。当最坏情况时，其时间复杂度为 $O(n^2)$ ，空间复杂度为 $O(1)$

(4) 堆排序所需的辅助空间比快排少，但这两种方法都不稳定。

(5) 归并排序既可以用于内部排序，也可以用于外部排序，是一种稳定的算法。

8、能熟练写出二分查找的程序。

9、熟悉算法的思想：贪心算法，动态规划，分治算法。

参考：<https://www.cnblogs.com/houjun/p/4896268.html>