1. 线性表顺序查找成功的**平均**长度为：(n+1)/2，查找不成功的**平均**长度为n+1
2. 有序表顺序查找查找成功的**平均**长度为：(n+1)/2，查找不成功的**平均**长度为n/2+2/(n+1)
3. 折半查找成功的**平均查找次数**为log2(n+1)-1，**最大查找次数**为树高 **向上取整log2(n+1)**，**最小**的为上式减1，因为判定树分支高度之差为1
4. 对于折半查找，若要求平均查找长度，则要画查找树
5. 查找判定树是一颗二叉排序树
6. 散列函数得选取方法：
    - 直接定值法：就是一个线性函数a*key+b
    - 除留余数法：取一个不大于散列表长，但是最接近散列表长的**质数p。**
    - 平方取中法：取关键字的平方值得中间几位作为散列地址
7. 散列冲突的处理方法：
    - 开放定址法：可存放新表项的空闲地址即向它的同义词表项开放，又向它的非同义词表项开放
        - 线性探查法：冲突发生时查看下一个空闲单元，会使得大量元素堆积起来，降低了查找效率
        - 平方探测法：可以避免堆积，但是不能探测到散列表上得所有单元
        - 再散列法：两个散列函数
    - 拉链法：
8. 散列表的平均查找长度依赖于散列表的装填因子a，而不直接依赖于n或m，a越大，装填的越满，发生冲突的可能性越大
9. 插入排序
    - 直接插入排序:是一个稳定的算法，适用于基本有序的排序表和数据量不大的排序表
    - 折半插入排序：折半排序的元素比较次数为nlogn，但是时间复杂度仍然是n^2
    - 希尔排序：再在最坏的情况下复杂度约为$n^{2,希尔排序的时间复杂度约为n}1.3$
10. 交换排序
    - 冒泡排序
    - 快速排序，牛逼o
11. 排序算法的稳定性：
    - 稳定算法：插入排序，冒泡排序，快速排序，归并排序，基数排序
    - 不稳定算法：直接选择排序，堆排序，
12. 初始堆的时间复杂度是N，堆排序的时间复杂度是nlogn
13. 快速排序的空间复杂度主要是递归，平均为logn，最坏为n
14. 主对角线下的元素均为0的话就说明了有向图是不存在环的，所以拓扑序列必然是存在的
15. 数据基本有序的时候，最佳的内部排序方法是**插入排序**
16. 哈夫曼树中所有**叶节点**带权路径长度之和为树的带权路径长度
17. 在含哟n个关键字的最小堆中，关键字最大的记录哟可能存储在[n/2] + 2 的位置上。
18. **栈**与数据的存储结构无关
19. 稀疏矩阵常用得两种存储方法是**三元组**和**十字链表**

20. 关键路径并不唯一，只提高一个关键路径上的活动不能加快，只哟提高包括在所有关键路径上的关键活动才能达到缩短工期的目的
21. 关键路径是指AOE网中的从开始顶点到结束顶点的最长路径
22. 需要分配一个较大的存储空间并且插入和删除操作不需要移动元素，满足以上特点的线性表存储结构是**静态链表**
23. 在有N个叶子结点的哈夫曼树中，其结点总数2N-1
24.