1. 随机选轴点，或者在排序之前进行shuffle操作

时间复杂度O（nlogN）最坏O(n^2)

空间复杂度(logN) 最坏(n)

<https://github.com/TheAlgorithms/Python/tree/master/sorts>

2. MergeSort

时间 O(nlogN)

空间 O(nlogN)

3. 看具体数据而定。

1）如果对排序稳定性要求选归并。

2）如果对稳定性要求不高，但对空间要求较高使用快排

<https://my.oschina.net/mjRao/blog/50669>

4.

1, logN, nlogN, n^2, n^3, 2^n, 2^2^n+1,

5. f(n) = n n^log3 3 = n T(n) = nlogn

f(n) = n^2.5 n^log2 4 = n^2 T(n) = f(n)

<https://blog.csdn.net/qq_35649707/article/details/77770463>

6. O(N)

7. heap, partition dive and conquer

8. newton iteration method,

mean method

binary search

9. partition

10. 动态数组， 链表

查找O(n), 查找O(n)

尾部增删O(1) O(1)

随机增删O(n) O(1)

11. 一种数据结构，也叫散列表，讲数据隐射到制定的空间范围内，O(1)的查找

空间复杂度O(n), collision就是hashi碰撞，

增加linklist

linear probe， quadratic probe等

.哈希表相对于其他数据结构的优缺点

1）优点：

记录数据量很大的时候，处理记录的速度很快，平均操作时间是一个不太大的常数

2）缺点：

①好的哈希函数（good hash function）的计算成本有可能会显著高于线性表或者搜索树在查找时的内部循环成本，所以当数据量非常小的时候，哈希表是低效的

②哈希表按照 key 对 value 有序枚举（ordered enumeration， 或者称有序遍历）是比较麻烦的（比如：相比于有序搜索树），需要先取出所有记录再进行额外的排序

③哈希表处理冲突的机制本身可能就是一个缺陷，攻击者可以通过精心构造数据，来实现处理冲突的最坏情况。即：每次都出现冲突，甚至每次都出现多次冲突（针对封闭散列的探测），以此来大幅度降低哈希表的性能。

12. nlogn 二分查找 + 遍历

\* 13.

我看了下短链接算法一些资料，大致有以下两种算法：

1、自增序列算法 也叫永不重复算法

设置 id 自增，一个 10进制 id 对应一个 62进制的数值，1对1，也就不会出现重复的情况。这个利用的就是低进制转化为高进制时，字符数会减少的特性。

短址的长度一般设为 6 位，而每一位是由 [a - z, A - Z, 0 - 9] 总共 62 个字母组成的，所以 6 位的话，总共会有 62^6 ~= 568亿种组合，基本上够用了。

算法二：

将长网址 md5 生成 32 位签名串,分为 4 段, 每段 8 个字节

对这四段循环处理, 取 8 个字节, 将他看成 16 进制串与 0x3fffffff(30位1) 与操作, 即超过 30 位的忽略处理

这 30 位分成 6 段, 每 5 位的数字作为字母表的索引取得特定字符, 依次进行获得 6 位字符串

总的 md5 串可以获得 4 个 6 位串,取里面的任意一个就可作为这个长 url 的短 url 地址

具体资料大家可以看下：

<https://hufangyun.com/2017/short-url/>

<https://www.zhihu.com/question/29270034>

<https://blog.csdn.net/wanghuan203/article/details/44775561>

14. B-Tree 因为B-Tree是一种外部数据结构，不需要全部加装入内存。而可以存储在外部存储器如HDD上

B树是一种树状数据结构，能够用来存储排序后的数据。这种数据结构能够让查找数据、循序存取、插入数据及删除的动作，都在对数时间内完成。B树，概括来说是一个一般化的二叉查找树，可以拥有多于2个子节点。与自平衡二叉查找树不同，B-树为系统最优化大块数据的读和写操作。B-tree算法减少定位记录时所经历的中间过程，从而加快存取速度。这种数据结构常被应用在数据库和文件系统的实作上。

因为B树每个node有大量的孩子，所以深度低于二叉树 。而且B树还能维护平衡 。适合存储在HDD中。一次遍历的HDD读写次数是logk N ， k是B树节点的孩子树

15. dynamic program

16. dynamic program

17. backtracking

18.

他俩最本质的区别是，贪心算法是具有贪心选择的性质的，而动态规划是含有重叠子问题性质的，可以理解成贪心算法是一种比较特殊的动态规划，由于其具有贪心选择性质，保证了子问题只会被计算一次，不会被多次计算

19. Hamming Distance

在信息论中，两个等长字符串之间的汉明距离是两个字符串对应位置的不同字符的个数。换句话说，它就是将一个字符串变换成另外一个字符串所需要替换的字符个数。

20. Union-find

21.