排序算法：

1. 冒泡排序（Bubble sort）

**先把最大的排最后；不断交换相邻两个元素**

1）假设序列共n个元素；

2）选取前两个元素，如果前一个元素大于后一个元素，则交换两个选择位置，则保持不变；继续选择第二、三个元素，如果前一个元素大于后一个，则交换位置，否则保持不变，依此类推，直到遍历完序列；

3）重复第二步操作；第m次遍历只遍历前n-m+1个元素（经过m-1次循环后，最后m-1个元素已经排序完成）

4）截止条件，如果一次遍历中没有任何位置调整，则截止；

<https://www.runoob.com/w3cnote/bubble-sort.html>

1. 选择排序（Selection Sort）

**先把最小的排最前；不断交换相邻两个元素**

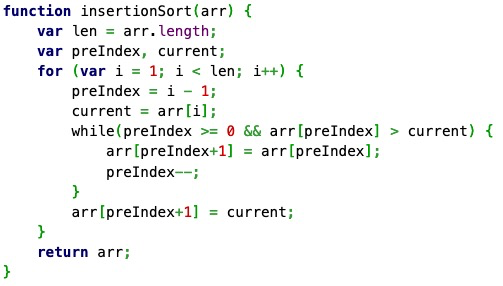
1. 假设序列共n个元素；
2. 遍历所有元素，选择其中最小的一个和第一个元素交换位置；
3. 重复第二步操作；第m次遍历只遍历后n-m+1个元素，将后n-m+1个元素中最小的元素与第m个元素交换位置
4. 截止条件，如果一次遍历中没有任何位置调整，则截止。

<https://www.runoob.com/w3cnote/selection-sort.html>

1. 插入排序（Insertion Sort）

**不断把后面最小的插入到前面有序序列（顺移）；**

1. 假设序列共n个元素；
2. 选择前两个元素，如果第2个元素大于第1个元素，则交换它们的位置
3. 重复第二步，选择第m个元素，如果它小于其前面的元素，则将其前面所有大于它的元素顺序后移一个位置，将第m个元素移动到本次未移动元素后面；否则保持不变；
4. 截止条件，如果未发生任何移动，则截止；



<https://www.runoob.com/w3cnote/insertion-sort.html>

1. 希尔排序（Shell Sort）

希尔排序是基于**插入排序**的以下两点性质而提出改进方法的：

插入排序在对几乎已经排好序的数据操作时，效率高，即可以达到**线性排序**的效率。

但**插入排序**一般来说是低效的，因为插入排序每次只能将数据移动一位（**插入时顺序移动给出空位**）。

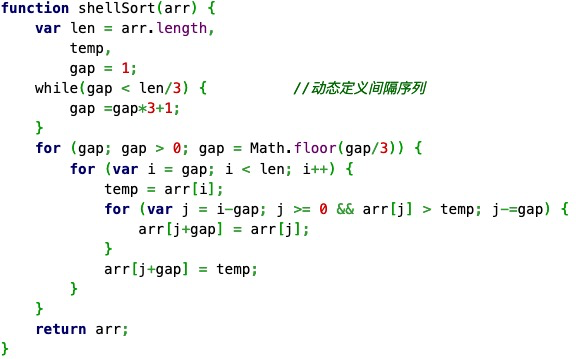
1. 假设序列共n个元素；

选择一个增量序列，，……，，其中 > , ；

按增量序列个数 ，对序列进行趟排序；

每趟排序，根据对应的增量，将待排序列分割成若干长度为的子序列，分别对各子表进行直接插入排序。仅增量因子为 1 时，整个序列作为一个表来处理，表长度即为整个序列的长度。

每趟排序处理元序列的一个子序列，即间隔为的所有元素组成一个子序列，最后一个增量对应的子序列为原序列，退化为普通插入排序。



<https://baike.baidu.com/item/%E5%B8%8C%E5%B0%94%E6%8E%92%E5%BA%8F>

<https://www.runoob.com/w3cnote/shell-sort.html>

1. 归并排序（Merge Sort）

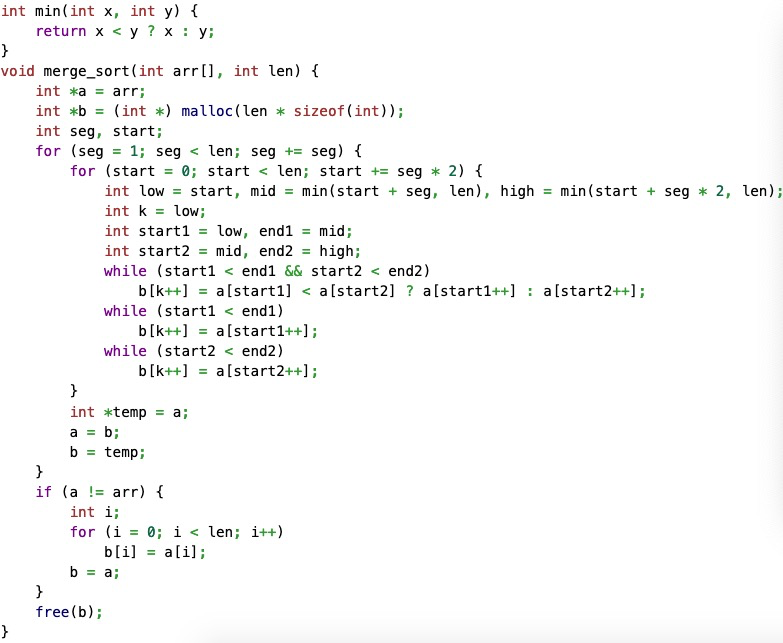
递归算法

1）将输入序列切分为两个子序列；分别对两个子序列排序（重复步骤1），并归并为一个子序列；

2）如果输入子序列元素数小于2，则立刻得到排序结果并返回；

或者

1. 将序列切分为元素数2的分组，依次对分组行排序；
2. 将排序的序列切分为元素数为的分组，依次对相邻分组进行排序到另一个序列。
3. 切换序列，依此执行步骤2，直至分组元素数大于等于序列长度。



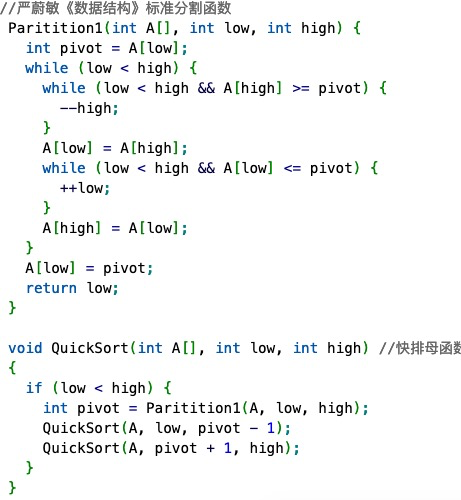
<https://www.runoob.com/w3cnote/merge-sort.html>

1. 快速排序（Quick Sort）

相比于归并排序，快速排序从大到小先排子序列长度大的序列；

而归并则先排列元素数最小的子序列，然后归并，再排列元素数大的归并结果子序列，直到完成排序；

1. 选定一个值，作为切分序列的标准；
2. 将大于标准值的元素排到右边，小于标准值的元素排左边，直至无元素移动；则原序列形成了两部分；
3. 再重新选定标准值，分别对左右两部分序列进行排序，依此类推，直到子序列无法再分，则排序完成。



<https://www.runoob.com/w3cnote/quick-sort-2.html>

1. 堆排序

类似完全二叉树

大顶堆：每个节点值都小于等于其父节点值；

小顶堆：每个节点值都大于等于其父节点值。

<https://www.runoob.com/w3cnote/heap-sort.html>

1. 计数排序

适合整数；用来计数的数组C的长度取决于待排数组中数据的范围，这使得它不适合排序范围很大的数。

记录每个数i出现的次数，并将这个数保存在目标数组第i个位置；

依次将目标数组的下标（原始数据）排列回原始序列，排列的位置为到，非常适合给学生排分数啊！！！

<https://www.runoob.com/w3cnote/counting-sort.html>

1. 桶排序

计数排序的升级版本。原来计数数组升级为桶位数组，每个桶位代表一个数值范围，即桶位就是排序好的。而在桶位对其中的原序进行独立排序。

<https://www.runoob.com/w3cnote/bucket-sort.html>

1. 基数排序

类比计数排序和桶排序，基数排序按照元素所处范围进行独立排序，如10-99，100-1000，切分数据

<https://www.runoob.com/w3cnote/radix-sort.html>