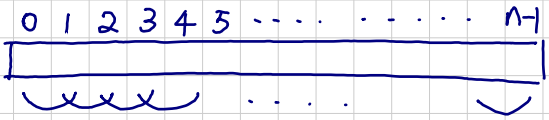
冒泡排序 Bubble Sort

时间复杂度 $O(n^2)$　　空间复杂度 $O(1)$ ← 因为只用了一个temp变量用于交换

n个元素排序，比较元素的下标是 [i] 与 [i+1]，从0开始范围是 0～n-2



i=0

第一次遍历，n个数，两两比较，共有 n-1 对数需要比较 [i] 与 [i+1]

所以第一次循环的范围是 0～n-2（共n-1对）

第一次遍历后，最大（或最小）的数就被交换到最后

第二次遍历，对 n-1（共 n-2 对）个数进行比较

⋮

i=n-2

第 n-1 次遍历，对 2（共1对）个数进行比较

遍历	比较次数
1	n-1
2	n-2
3	n-3
···	···
n-1	1

! 重点　　时间 $O(n^2)$　空间 $O(1)$

基本操作：遍历

比较 [i] 与 [i+1]

交换 [i] 与 [i+1]

选择排序（selection sort）

选择排序优化了冒泡排序，每一次遍历只做一次交换。

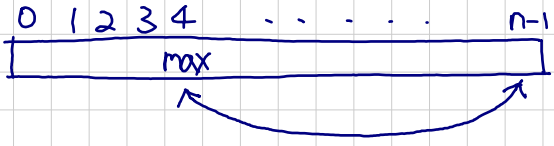
时间复杂度 $O(n^2)$，空间复杂度 $O(1)$ ← 因为只是多了一个 pos_max 变量记录每次遍历中的最大值。

对于长度为 n 的数组：

同样，选择排序一共有 n-1 遍历

- 需要一个变量记录最大值的位置
- 每一次遍历，最大值的位置都被初始化为 0 （这个不保证是最左或最右就好了）

第一次遍历，把最大值交换到最后



❗重点　　时间 $O(n^2)$　空间 $O(1)$

基本操作：遍历

比较：pos_max 与这一次遍历中所有其他元素

交换：pos_max 与这一次遍历中最后一个元素（把最大的交换到后面）

插入排序（insertion sort）

时间复杂度 $O(n^2)$，空间复杂度 $O(1)$

插入排序总是假设左边的 sublist 已经排好序了，目的是把当前元素插入到已排好序的 sublist 中。

对于长度为 n 的数组：

- 假设第 0 个元素已经排好序，所以循环是从 1（第2个元素） ～ n-1（最后一个元素）
- 每一次遍历，都需要两个变量：保存当前的值 current_value；保存当前的位置 index
- 基本操作，当当前的位置大于零（因为开始时假设位置零的元素是排过序的），且当前的值（current_value）小于当前值前面一位的值（alist[position-1]），前一位的值往右移一位

Current_value

1 4 2 7　　2 < 4 ? Yes

1 ? 4 7

insert or continue find the proper position?

alist[position] = alist[position-1]

如果不满足条件，则

alist[position] = current_value

To be continued °°°

插入排序，continue

❗ 重点： 时间复杂度 $O(n^2)$，空间复杂度 $O(1)$

基本操作：遍历：（假设位置0已排序，
遍历从1开始到最后一个元素）

比较：当前元素是否小于前一位的值

移动：满足条件后，{前一位元素右移
当前位置左移

插入：找到合适位置，再插入
（之前的操作
只有左/右移）

希尔排序（Shell sort）