Python插入排序

讲师：Wayne

从业十余载，漫漫求知路

直接插入排序

- 直接插入排序原理
  - 在未排序序列中，构建一个子排序序列，直至全部数据排序完成
  - 将待排序的数，插入到**已经排序**的序列中合适的位置
  - 增加一个哨兵，放入待比较值，让它和后面已经排好序的序列比较，找到合适的插入点

直接插入排序Direct insertion sort

- 初始 0 1 9 8 5 6
- 第一趟 9 1 9 8 5 6
- 第二趟 8 1 9 8 5 6 | 8 1 ?→9 5 6 | 8 1 8 9 5 6
- 第三趟 5 1 8 9 5 6 | 5 1 8 9→9 6 | 5 1 ?→8 9 6 | 5 1 5 8 9 6
- 第四趟 6 1 5 8 9 6 | 6 1 5 8 9→9 | 6 1 5 ?→8 9 | 6 1 5 6 8 9
- 开头的红色数字为哨兵，即待插入值
- 假定1已经有序，从第二个数字9开始排序
- 第一趟，哨兵9，1和哨兵比较，1小，本轮比较结束
- 第二趟，哨兵8，9和哨兵比较，哨兵9大右移，1和哨兵比较，1小，哨兵插入本轮比较结束
- 以此类推，直至把最后一个数字放到哨兵并比较、插入完成

直接插入排序

- 增加一个哨兵位，每轮比较将待比较数放入
- 哨兵依次和待比较数的前一个数据比较，大数靠右移动，找到哨兵中值的插入位置
- 每一轮结束后，得到一个从开始到待比较数位置的一个有序序列

```
m_list = [
    [1, 9, 8, 5, 6, 7, 4, 3, 2], [1, 2, 3, 4, 5, 6, 7, 8, 9],
    [9, 8, 7, 6, 5, 4, 3, 2, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1],
    [1, 1, 1, 1, 1, 1, 1, 1, 2]]
#nums = [0, 1, 9, 8, 5, 6]
nums = [0] + m_list[0]
sentinel, *origin = nums   #哨兵位,待比较数字

count_swap = 0
count_iter = 0

length = len(nums)
for i in range(2,length): # 从2开始
    nums[0] = nums[i] # 放置哨兵
    j = i - 1 #
    count_iter += 1
    if nums[j] > nums[0]: # 大数右移，找到插入位置
        while nums[j] > nums[0]:
            nums[j+1] = nums[j] # 依次右移
            j -= 1
            count_swap += 1
        nums[j+1] = nums[0] # 将哨兵插入，注意插入在右侧要+1
print(nums, count_swap, count_iter)
```

直接插入排序

- 最好情况，正好是升序排列，比较迭代n-1次
- 最差情况，正好是降序排列，比较迭代1,2,...,n-1即 n(n-1)/2，数据移动非常多
- 使用两层嵌套循环，时间复杂度O(n^2)
- 稳定排序算法
- 使用在小规模数据比较
- 优化
  - 如果比较操作耗时大的话，可以采用二分查找来提高效率，即二分查找插入排序

谢谢

咨询热线 400-080-6560