



问题解答

排序算法要点，稳定排序

慕课测验详解

关于期末和大作业

数据结构与算法 (Python) -12/0407

陈斌 gischen@pku.edu.cn 北京大学地球与空间科学学院

目录

- › 问题解答
- › 各种排序算法的要点
- › 什么是稳定的排序？
- › 慕课测验详解
- › 关于期末大作业
规则 and 分组
- › 课堂练习



问题解答

- › 请问是不是所有用到递归和动态规划的都应该叫体现了分治思想？或者说大部分？因为动规和递归都是利用更小规模的相同问题来解决整个问题。
- › 为什么还有学习选择排序、冒泡排序的必要呢？
- › 为什么快速排序法不能用限制条件使它时间复杂度变为 $O(N)$ ？
- › 希尔排序问题，考试可以回答希尔排序的最好版本吗？（ $O(N^{3/2})$ 的）
- › 什么是稳定排序？
- › 如何理解排序稳定性？不稳定的算法劣势在哪里？

问题解答

- › 归并排序不能用递归做吗？
- › 归并是不是可以稳定也可以不稳定？
- › 睡眠排序

问题解答

- › 大作业胜利条件是否确定？
- › 有期末题型吗？
- › 老师怎么做到用那个搜索引擎的？每次一打开浏览器自动连接吗？
- › ****C1获奖作品**

本周内容：排序与查找（上）

- › 501 顺序查找算法及分析 9m41s
- › 502 二分查找算法及分析 12m20s
- › 503 冒泡和选择排序算法及分析 12m14s
- › 504 插入排序算法及分析 7m06s
- › 505 谢尔排序算法及分析 6m15s
- › 506 归并排序算法及分析 9m13s
- › 507 快速排序算法及分析 12m30s

各种排序的要点

- › **冒泡排序**：最稳定 n^2 ；不需要额外空间；
- › **选择排序**：同冒泡，记录最大值，只做1次交换；
- › **插入排序**：平均 n^2 ，最好 n ；不需要额外空间；
从1项开始，逐步扩大排序子表，寻找“新项”插入位置
在已排序子表中，从后往前，比对、移动所有比“新项”大的数据项
- › **谢尔排序**：在 n 、 n^2 之间；不需要额外空间；
在子列表中采用插入排序gapInsertionSort
从 $n/2$ 倍增减小gap直到1，子列表个数直到1个，排好序

各种排序的要点

› **归并排序：稳定 $n\log n$ ；需要1倍额外空间；**

以“**中间位置**”作为分裂**子列表**的依据；

递归->合并

› **快速排序：最好、平均 $n\log n$ ；最差 n^2 ；不需要额外空间；**

以“**中间值**”作为分裂**子列表**的依据；

分裂->递归

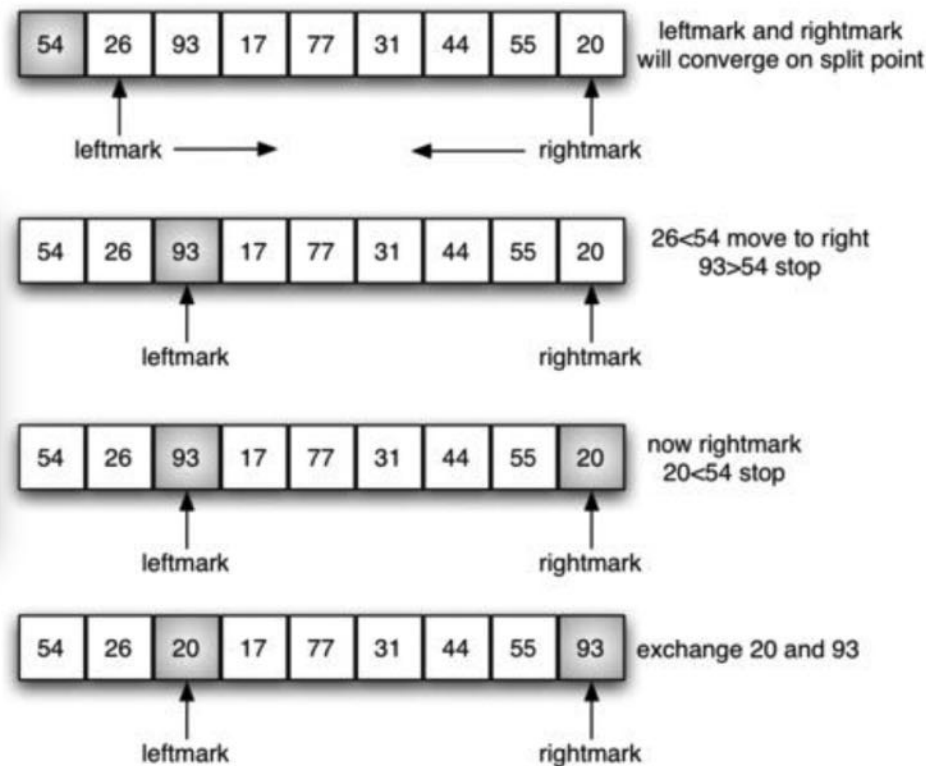
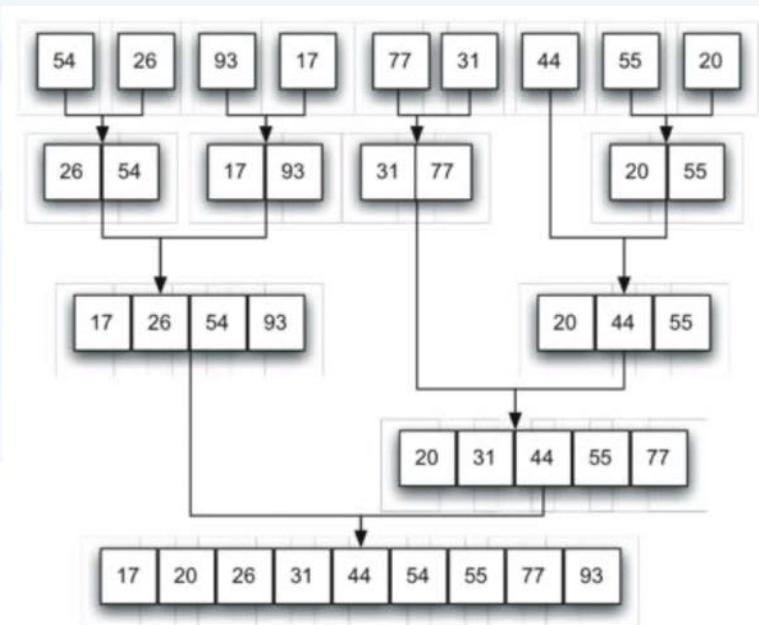
数据的初始分布、取中间值的方法特别重要

什么是稳定的排序？

- › **保证排序前2个相等数的前后位置，和排序后前后位置顺序相同**
稳定性的好处：支持多个键值排序；可以减少一些交换次数；
- › **冒泡排序是稳定的：交换只发生在相邻两个数据项之间；**
- › **选择排序不稳定：可能会跨越多个数据项交换；**
- › **插入排序是稳定的：从后向前寻找插入位置，不改变相对；**
- › **谢尔排序不稳定：虽然以插入排序为基础，但多个子表是穿插的**
- › **基数排序是稳定的：相同数据项始终按顺序进入队列，出队列**

什么是稳定的排序？

- 归并排序是稳定的：在合并的过程中，左半部始终在前面，也不破坏相同数据项的相对位置
- 快速排序不稳定：在右标左移的过程中，会把尾部的数，跳跃交换到前面，破坏相对位置



另一些奇奇怪怪的排序

› 睡眠排序 (Sleep Sort)

把每个数对应一个线程，同时启动这些线程

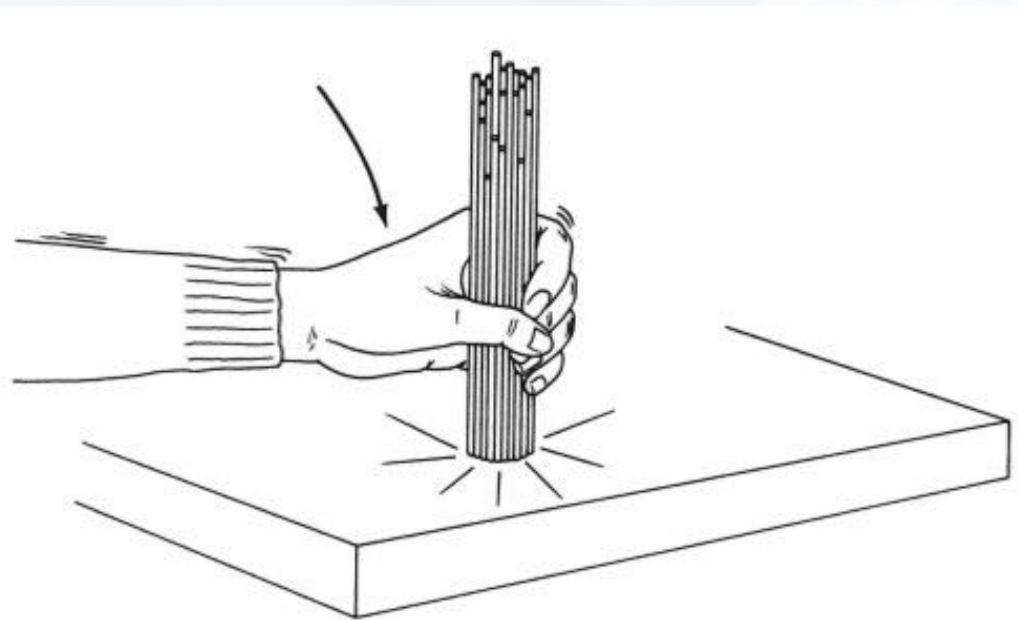
每个线程都是sleep对应数的时间，然后输出这个数

完成排序

```
1 import _thread
2 from time import sleep
3
4 items = [6, 2, 5, 3, 1, 7]
5
6 def sleep_sort(i):
7     sleep(i)
8     print(i)
9
10 [_thread.start_new_thread(sleep_sort, (i,)) for i in items]
```

另一些奇奇怪怪的排序

面条排序(Spaghetti Sort)



另一些奇奇怪怪的排序

› 猴子排序 (Bogo Sort)

```
1  from random import shuffle
2
3  def inorder(lst):
4      for i in range(len(lst) - 1):
5          if lst[i] > lst[i + 1]:
6              return False
7      return True
8
9  def bogo(x):
10     while not inorder(x):
11         shuffle(x)
12     return x
13
14  l = bogo([1,3,2,4,5])
15  print(l)
```

慕课测验详解（OJ作业参见助教文档）



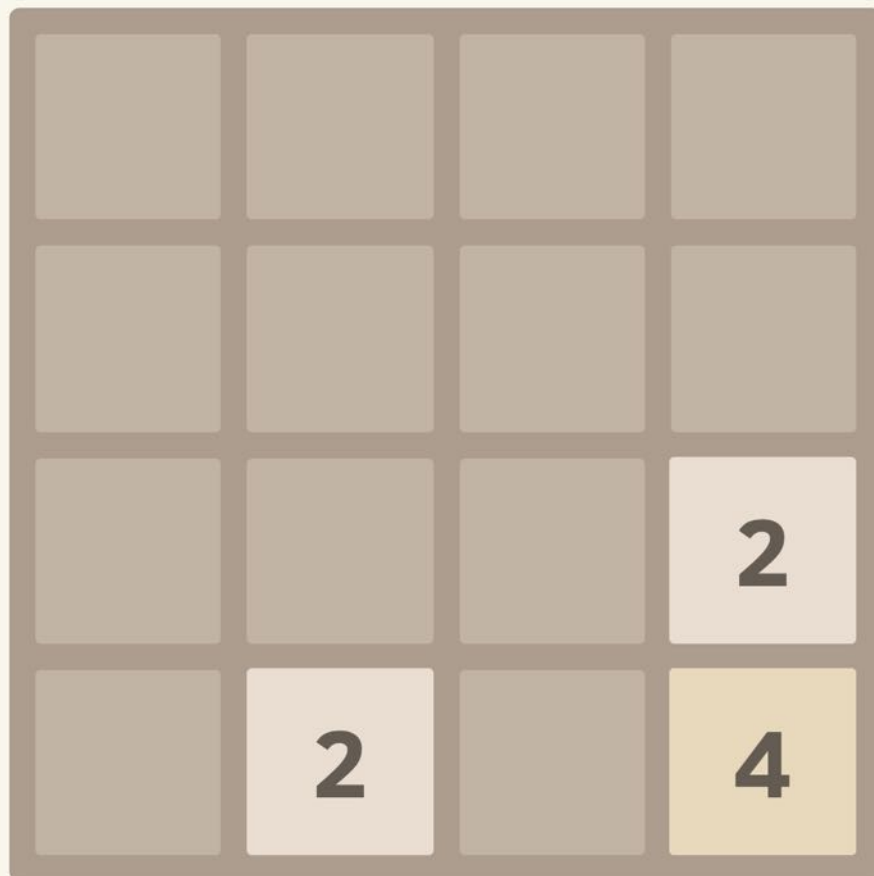
期末考试？

› 太着急了~

› 往年试卷：

› <http://gis4g.pku.edu.cn/course/pythonds-2019/>

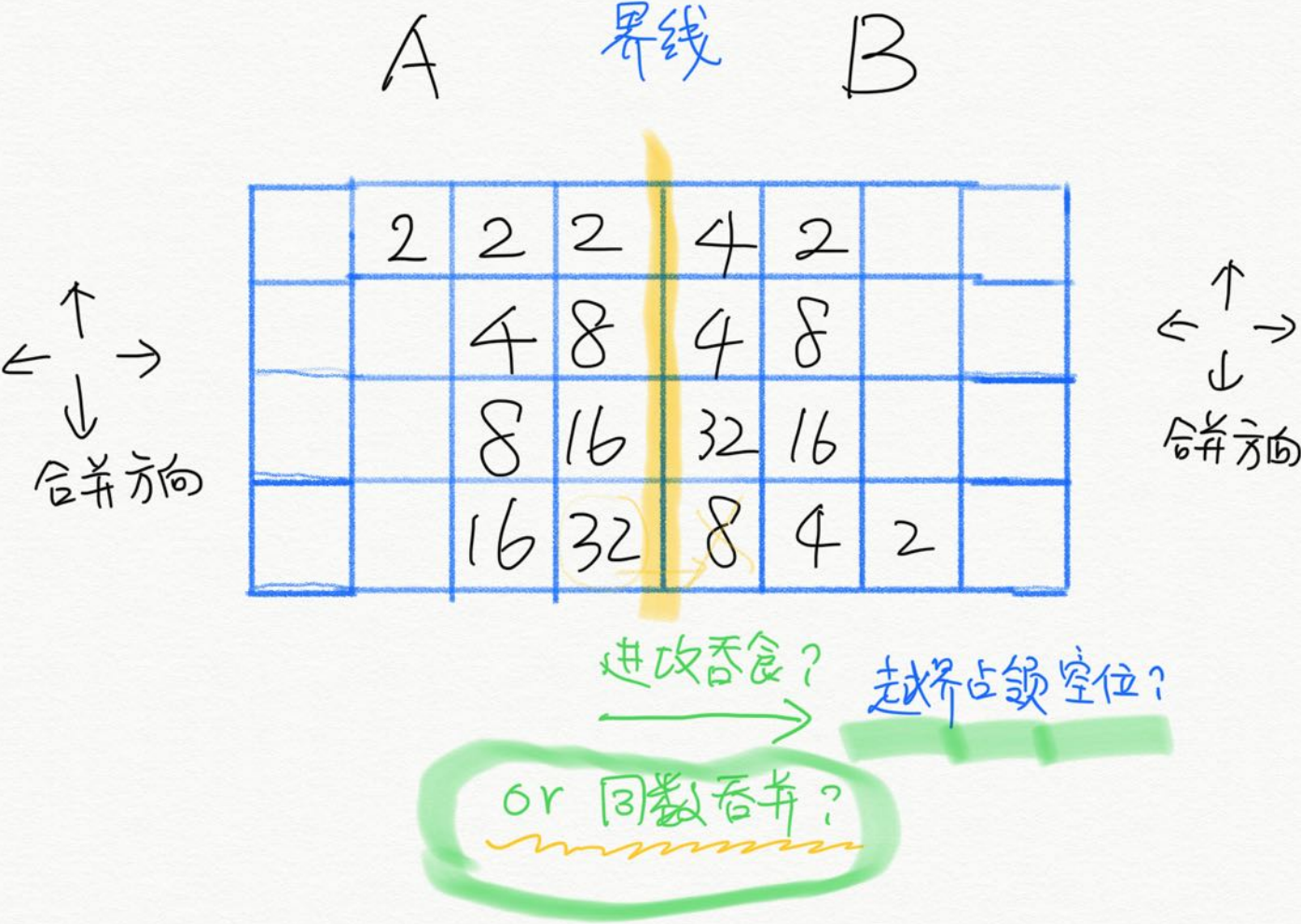
2048 (<https://play2048.co/>)



HOW TO PLAY: Use your **arrow keys** to move the tiles. When two tiles with the same number touch, they **merge into one!**

期末大作业 (sessdsa.2048)

数据结构与算法 (Python)



课堂练习

- › 5道选择题，在canvas上提交。