

# 作业六：排序算法类的实现

陈俊铭 3210300364 （信息与计算科学）

2022 年 11 月 25 日

## 1 设计思路

使用和比较两种排序算法，快速排序 (quicksort) 和堆排序 (heapsort)。

## 2 测试结果

When  $n = 10000$ ,

- efficiency: 0.01, heaptime: 0.003 s, quicktime: 0 s.
- efficiency: 0.1, heaptime: 0.006 s, quicktime: 0.001 s.
- efficiency: 0.9, heaptime: 0.005 s, quicktime: 0.001 s.
- efficiency: 0.99, heaptime: 0.003 s, quicktime: 0.001 s.

When  $n = 100000$ ,

- efficiency: 0.01, heaptime: 0.035 s, quicktime: 0.005 s.
- efficiency: 0.1, heaptime: 0.028 s, quicktime: 0.005 s.
- efficiency: 0.9, heaptime: 0.022 s, quicktime: 0.006 s.
- efficiency: 0.99, heaptime: 0.034 s, quicktime: 0.006 s.

When  $n = 1000000$ ,

- efficiency: 0.01, heaptime: 0.286 s, quicktime: 0.065 s.
- efficiency: 0.1, heaptime: 0.284 s, quicktime: 0.053 s.
- efficiency: 0.9, heaptime: 0.274 s, quicktime: 0.051 s.
- efficiency: 0.99, heaptime: 0.27 s, quicktime: 0.056 s.

### 3 快排改进

当输入包含太多相同元素时，我们可以对其进行润色。选择主元，或围绕其划分列表的元素，是快速排序中的关键操作之一。快速排序中的枢轴元素通常是分区的最外层元素之一，要么在左边，要么在右边。对于已排序或接近已排序的输入，此选择将导致最坏的情况。通过选择主元分区的第一个、中间的和最后一个元素的中位数，或者为主元选择一个随机索引，我们可以快速高效地解决问题（median3）。