

# Heapsort 与 Quicksort 之比较

廖玺然

Nanjing University

May 21, 2018

# Theoretical

Quicksort:

# Theoretical

Quicksort:

$$C(n) = n - 1 + \frac{1}{n} \sum_{i=0}^{n-1} (C(i) + C(n - i - 1)) = 2n \ln n = 1.39n \log_2 n$$

# Theoretical

Quicksort:

$$C(n) = n - 1 + \frac{1}{n} \sum_{i=0}^{n-1} (C(i) + C(n - i - 1)) = 2n \ln n = 1.39n \log_2 n$$

Heapsort: ?

# Theoretical

cache

“快排几十年前就被发现多数情况比堆排快了，而那时候的 cpu 除了寄存器基本没 cache”

# Theoretical

cache

“快排几十年前就被发现多数情况比堆排快了，而那时候的 cpu 除了寄存器基本没 cache”

“快排产生的指令数量开销比堆排产生的指令数量开销小”

# Experimental



# Application



# Application

Quicksort: 数据量大, 离散程度高

# Application

Quicksort: 数据量大, 离散程度高

Heapsort:

# Application

Quicksort: 数据量大, 离散程度高

Heapsort:

1. 在 100 万个数中查找最大的 100 个

# Application

Quicksort: 数据量大, 离散程度高

Heapsort:

1. 在 100 万个数中查找最大的 100 个
2. 对一个基本有序的数组进行排序

# Application

Quicksort: 数据量大, 离散程度高

Heapsort:

1. 在 100 万个数中查找最大的 100 个
2. 对一个基本有序的数组进行排序
3. 空间复杂度为  $O(1)$ , 判断数组中是否有重复值

# Reference

*[https://blog.csdn.net/qq\\_28352347/article/details/69159271](https://blog.csdn.net/qq_28352347/article/details/69159271)*

*<https://blog.csdn.net/shakespeare001/article/details/51360732>*

*<https://www.zhihu.com/question/23873747>*

