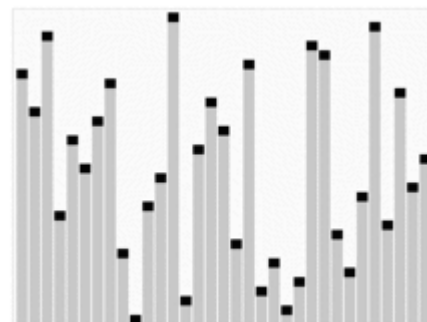


# Пирамида́льная сорти́ровка

Материал из Википедии — свободной энциклопедии

**Пирамида́льная сорти́ровка** (англ. *Heapsort*, «Сортировка кучей»<sup>[1]</sup>) — алгоритм сортировки, работающий в худшем, в среднем и в лучшем случае (то есть гарантированно) за  $\Theta(n \log n)$  операций при сортировке  $n$  элементов.<sup>[2]</sup> Количество применяемой служебной памяти не зависит от размера массива (то есть,  $O(1)$ ).

Может рассматриваться как усовершенствованная сортировка пузырьком, в которой элемент всплывает (min-heap) / тонет (max-heap) по многим путям.



Анимированная схема алгоритма

## Содержание

История создания

Алгоритм

Достоинства и недостатки

Применение

Примечания

Литература

Ссылки

## История создания

Пирамида́льная сорти́ровка была предложена Дж. Уильямсом в 1964 году.<sup>[1]</sup>

## Алгоритм

Сортировка пирамидой использует бинарное сортирующее дерево.

Сортирующее дерево — это такое дерево, у которого выполнены условия:

- Каждый лист имеет глубину либо  $d$ , либо  $d - 1$ ,  $d$  — максимальная глубина дерева.
- Значение в любой вершине не меньше (другой вариант — не больше) значения её потомков.

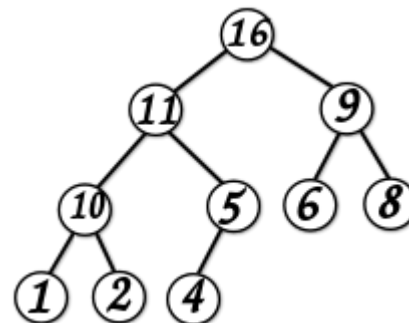
Удобная структура данных для сортирующего дерева — такой массив Array, что Array[0] — элемент в корне, а потомки элемента Array[i] являются Array[2i+1] и Array[2i+2].

Алгоритм сортировки будет состоять из двух основных шагов:

- Выстраиваем элементы массива в виде сортирующего дерева:

Array[*i*] ≥ Array[2*i* + 1]

Array[*i*] ≥ Array[2*i* + 2]



Пример сортирующего дерева

при  $0 \leq i < n/2$ .

Этот шаг требует  $O(n)$  операций.

16	11	9	10	5	6	8	1	2	4
----	----	---	----	---	---	---	---	---	---

структура хранения данных  
сортирующего дерева

2. Будем удалять элементы из корня по одному за раз и перестраивать дерево. То есть на первом шаге обмениваем `Array[0]` и `Array[n-1]`, преобразовываем `Array[0], Array[1], ..., Array[n-2]` в сортирующее дерево. Затем переставляем `Array[0]` и `Array[n-2]`, преобразовываем `Array[0], Array[1], ..., Array[n-3]` в сортирующее дерево. Процесс продолжается до тех пор, пока в сортирующем дереве не останется один элемент. Тогда `Array[0], Array[1], ..., Array[n-1]` — упорядоченная последовательность.

Этот шаг требует  $O(n \log n)$  операций.

## Достоинства и недостатки

### Достоинства

- Имеет доказанную оценку худшего случая  $O(n \cdot \log n)$ .
- Сортирует на месте, то есть требует всего  $O(1)$  дополнительной памяти (если дерево организовывать так, как показано выше).

### Недостатки

- Неустойчив — для обеспечения устойчивости нужно расширять ключ.
- На почти отсортированных массивах работает столь же долго, как и на хаотических данных.
- На одном шаге выборку приходится делать хаотично по всей длине массива — поэтому алгоритм плохо сочетается с кэшированием и подкачкой памяти.
- Методу требуется «мгновенный» прямой доступ; не работает на связанных списках и других структурах памяти последовательного доступа.

Сортировка слиянием при расходе памяти  $O(n)$  быстрее ( $O(n \cdot \log n)$  с меньшей константой) и не подвержена деградации на неудачных данных.

Из-за сложности алгоритма выигрыш получается только на больших  $n$ . На небольших  $n$  (до нескольких тысяч) быстрее сортировка Шелла.

## Применение

Алгоритм «сортировка кучей» активно применяется в ядре Linux (<http://lxr.free-electrons.com/source/lib/sort.c>).

## Примечания

1. Курс лекций «Современные технологии параллельного программирования», Лекция № 5: Пирамидальная сортировка (<http://ipro.ru/parallel-programming/lecture-5/>)
2. ScienceDirect — Journal of Algorithms : The Analysis of Heapsort (<https://dx.doi.org/10.1006/jagm.1993.1031>)

## Литература

- Левитин А. В. Глава 6. Метод преобразования: Пирамиды и пирамидальная сортировка // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 275–284. — 576 с. — ISBN 978-5-8459-0987-9
- Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Глава 6. Пирамидальная сортировка // Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. — 2-е изд. — М.: Вильямс, 2005. — С. 182—188. — ISBN 5-8459-0857-4.

## Ссылки

- [Пирамидальная сортировка \(http://iproc.ru/parallel-programming/lection-5/\)](http://iproc.ru/parallel-programming/lection-5/) — подробное описание с иллюстрациями и примером реализации на C++. Приведён вывод оценок скорости работы алгоритма и измерение времени работы на реальной вычислительной системе.
- [Сортировка с помощью кучи \(пирамидальная сортировка\) \(https://web.archive.org/web/20090926194051/http://p-analiz.jino-net.ru/sort\\_heap.htm\)](https://web.archive.org/web/20090926194051/http://p-analiz.jino-net.ru/sort_heap.htm) — доходчивое описание с иллюстрациями и примером реализации на Pascal.
- [Динамическая визуализация 7 алгоритмов сортировки с открытым исходным кодом \(https://airtucha.github.io/SortVis/\)](https://airtucha.github.io/SortVis/)

---

Источник — [https://ru.wikipedia.org/w/index.php?title=Пирамидальная\\_сортировка&oldid=98258539](https://ru.wikipedia.org/w/index.php?title=Пирамидальная_сортировка&oldid=98258539)

---

**Эта страница в последний раз была отредактирована 22 февраля 2019 в 03:46.**

Текст доступен по [лицензии Creative Commons Attribution-ShareAlike](#); в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации [Wikimedia Foundation, Inc.](#)