

# Плавная сортировка

Материал из Википедии — свободной энциклопедии

**Плавная сортировка** (англ. *Smoothsort*) — алгоритм сортировки выбором, разновидность пирамидальной сортировки, разработанная Э. Дейкстрой в 1981 году. Как и пирамидальная сортировка, имеет сложность в худшем случае равную  $O(n \log n)$ . Преимущество плавной сортировки в том, что её сложность приближается к  $O(n)$ , если входные данные частично отсортированы, в то время как у пирамидальной сортировки сложность всегда одна, независимо от состояния входных данных.

## Содержание

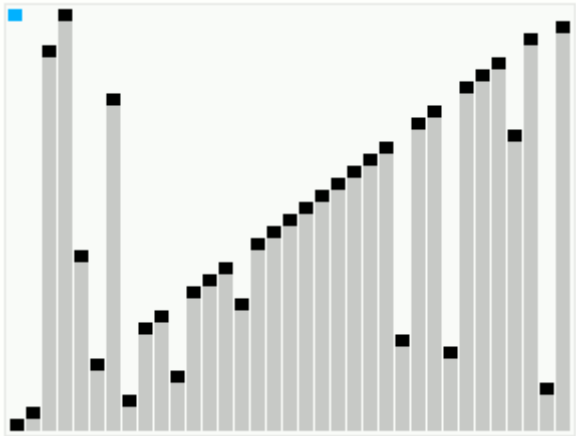
### Общее представление

#### Операции

- Увеличение последовательности куч путём добавления элемента справа
  - Оптимизация
- Уменьшение последовательности куч путём удаления элемента справа
  - Оптимизация

#### Используемая память

#### Ссылки



Ход плавной сортировки. Обрабатывается почти упорядоченный массив, отдельные элементы которого «выпадают» из последовательности

## Общее представление

Как и в пирамидальной сортировке, в массив накапливается *куча* из данных, которые затем сортируются путём непрерывного удаления максимума из кучи. В отличие от пирамидальной сортировки, здесь используется не *двоичная куча*, а специальная, полученная с помощью *чисел Леонардо*. Куча состоит из последовательности куч, размеры которых равны одному из чисел Леонардо, а корни хранятся в порядке возрастания. Преимущества таких специальных куч перед двоичными состоят в том, что если последовательность отсортирована, её создание и разрушение займёт  $O(n)$  времени, что будет быстрее. Разбить входные данные на кучи просто: крайние слева узлы массива составят самую большую кучу, а оставшиеся будут разделены подобным образом.

- Массив любой длины может быть также разбит на части размера  $L(x)$ .
- Не должно быть двух куч одинакового размера, потому что в этом случае последовательность превратится в строго убывающую по размерам.
- Не должно быть двух куч, размеры которых равны двум последовательным числам Леонардо, за исключением массива длины 2.

Эти положения могут быть доказаны.

Каждая куча размера  $L(x)$  состоит, слева направо, из подкучи размера  $L(x - 1)$ , подкучи размера  $L(x - 2)$  и корня, за исключением куч размера  $L(1)$  и  $L(0)$ , которые состоят только из корня. Для каждой кучи выполняется следующее свойство: значение корня должно быть не меньше значений корней его куч-потомков (и, как следствие, не меньше значений всех узлов его куч-потомков). Для последовательности куч в свою очередь выполняется следующее свойство: значение корня каждой кучи должно быть не меньше значения корня кучи слева. Следствие этого — крайний правый узел в последовательности будет иметь наибольшее значение, и, что важно, частично отсортированный массив не будет нуждаться в перестановке элементов, для того чтобы стать правильной последовательностью куч. Это является источником приспособляемости алгоритма. Алгоритм прост. Сначала происходит разделение неотсортированного массива на кучу с одним элементом и неотсортированную часть. Куча с одним элементом, очевидно, представляет собой правильную последовательность куч. Последовательность начинает расти путём добавления по одному элементу справа за раз (если нужно, элементы меняются местами, чтобы выполнялось свойство кучи и свойство последовательности), пока не достигнет размера изначального массива. И с этого момента крайний правый элемент в последовательности куч будет самым большим для любой кучи, а, следовательно, будет находиться на верной, конечной позиции. Затем последовательность куч уменьшается до кучи с одним элементом при помощи удаления крайнего правого узла и изменения позиций элементов для восстановления состояния кучи. Как только останется куча с одним элементом, массив будет отсортирован.

## Операции

Необходимы две операции: увеличение последовательности куч путём добавления элемента справа и уменьшение путём удаления крайнего правого элемента (корня последней кучи), с сохранением состояния кучи и последовательности куч.

### Увеличение последовательности куч путём добавления элемента справа

- Если две последние кучи имеют размеры  $L(x + 1)$  и  $L(x)$  (двух последовательных чисел Леонардо), новый элемент становится корнем кучи большего размера, равного  $L(x + 2)$ . Для неё свойство кучи необязательно.
- Если размеры двух последних куч не равны двум последовательным числам Леонардо, новый элемент образует новую кучу размером 1. Этот размер полагается равным  $L(1)$ , кроме случая, когда крайняя правая куча уже имеет размер  $L(1)$ , тогда размер новой одноэлементной кучи полагают равным  $L(0)$ .

После этого должно быть восстановлено выполнение свойств кучи и последовательности куч, что, как правило, достигается при помощи разновидности сортировки вставками:

1. Крайняя правая куча (сформированная последней) считается «текущей» кучей.
2. Пока слева от неё есть куча и значение её корня больше значения текущего корня и обоих корней куч-потомков:
  - Меняются местами новый корень и корень кучи слева (что не нарушит свойство для текущей кучи). Эта куча становится текущей.
3. Выполняется «просеивание», чтобы выполнялось свойство кучи:
  - Пока размер текущей кучи больше 1 и значение корня любой из куч-потомков больше значения корня текущей кучи:
    - Меняются местами наибольший по значению корень кучи-потомка и текущий корень. Куча-потомок становится текущей кучей.

Операция просеивания значительно упрощена благодаря использованию чисел Леонардо, так как каждая куча либо будет одноэлементной, либо будет иметь двух потомков. Нет нужды беспокоиться об отсутствии одной из куч-потомков.

### Оптимизация

- Если ожидается, что новая куча станет частью текущей, к моменту окончания, не нужно проверять выполнение свойства кучи: это понадобится только в случае, если куча достигнет конечного состояния.
  - Для этого нужно взглянуть на количество оставшихся после формирования кучи размера  $L(x)$  элементов. Если оно больше, чем  $L(x - 1) + 1$ , тогда эта новая куча будет поглощена.
- Необязательно соблюдать выполнение свойства кучи для крайней правой кучи. Если эта куча станет одной из конечных куч последовательности куч, то выполнение свойства последовательности куч обеспечит выполнение свойства кучи. Конечно, всякий раз когда формируется новая куча, крайняя правая куча становится не крайней правой, и выполнение свойства кучи должно быть восстановлено.

## Уменьшение последовательности куч путём удаления элемента справа

Если размер крайней правой кучи равен 1 — то есть это куча  $L(1)$  или  $L(0)$ , — то эта куча просто удаляется. В противном случае корень этой кучи удаляется, кучи-потомки считаются элементами последовательности куч, после чего проверяется выполнение свойства кучи, сначала для левой кучи, затем — для правой.

### Оптимизация

- Значения корня кучи слева и значения узлов куч, которые только что образовались из куч-потомков, не сравниваются, так как известно, что свойство для них выполняется. Поэтому сравнение происходит только со значением корня.

## Используемая память

Алгоритм плавной сортировки требует памяти для хранения размеров всех куч в последовательности. Так как все эти значения различны, как правило, для этой цели применяется битовая карта. Кроме того, так как в последовательности не больше  $O(\log n)$  чисел, биты могут быть закодированы  $O(1)$  машинными словами при условии использования трансдихотомической модели.

## Ссылки

- Плавная сортировка (Smooth\_sort) (<http://cppalgo.blogspot.ru/2010/10/smoothsort.html>)
- E. W. Dijkstra. Smoothsort — an alternative to sorting in situ (EWD-796a) (<http://www.cs.utexas.edu/users/EWD/ewd07xx/EWD796a.PDF>) (англ.)

Источник — [https://ru.wikipedia.org/w/index.php?title=Плавная\\_сортировка&oldid=87065007](https://ru.wikipedia.org/w/index.php?title=Плавная_сортировка&oldid=87065007)

**Эта страница в последний раз была отредактирована 14 августа 2017 в 08:32.**

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.