



Алгоритм сортировки кучей (Heap Sort)

Определение

Алгоритм сортировки кучей, также известный как пирамидальная сортировка (Heap Sort), относится к сортировкам на основе сравнения. Он использует структуру данных под названием "куча" для упорядочивания элементов в массиве. Куча является двоичным деревом, где каждый узел удовлетворяет условию "родитель \geq дети" или "родитель \leq дети" в зависимости от типа кучи (максимальная или минимальная).

Определение

Сценарий 1



Сценарий 2



Основные шаги алгоритма

1. Построение кучи: Начните с построения кучи из исходного массива. Один из способов построения кучи - это преобразование массива в бинарное дерево, удовлетворяющее свойствам кучи.

2. Превращение кучи в отсортированный массив: После построения кучи, наибольший (в случае максимальной кучи) или наименьший (в случае минимальной кучи) элемент массива находится в корне кучи. Этот элемент заменяется последним элементом массива, после чего размер кучи уменьшается на 1. Затем выполняется операция "просеивания вниз" (heapify), чтобы убедиться, что новый корень удовлетворяет свойствам кучи. Этот процесс повторяется до тех пор, пока весь массив не будет отсортирован.

3. Повторение шага 2 до полной сортировки: Продолжайте извлекать корень кучи и восстанавливать свойства кучи до тех пор, пока куча не опустеет, и весь массив не будет отсортирован.

Преимущества и недостатки

Преимущества алгоритма сортировки кучей включают в себя его стабильность и возможность применения для сортировки на месте (in-place sorting).

Однако, его основной недостаток - это более высокая сложность по сравнению с некоторыми другими алгоритмами сортировки, такими как быстрая сортировка (quicksort) или сортировка слиянием (merge sort).

Свойства

1. Временная сложность: В худшем и среднем случае время выполнения сортировки кучей составляет $O(n * \log_2 n)$, где n - количество элементов в массиве. Это гарантированное время выполнения, что делает алгоритм эффективным для больших объемов данных.

2. In-place сортировка: Алгоритм сортировки кучей работает на месте, то есть не требует дополнительной памяти за пределами исходного массива. Это позволяет сократить использование памяти и делает его привлекательным для сортировки больших объемов данных.

3. Неустойчивость: В общем случае, сортировка кучей является неустойчивой, то есть она не сохраняет порядок элементов с одинаковыми значениями. Однако, при необходимости, это свойство можно модифицировать для сохранения устойчивости.

Свойства

4. Отсутствие зависимости от исходного расположения элементов: Алгоритм сортировки кучей не зависит от начального расположения элементов в массиве. Он обеспечивает стабильное время выполнения независимо от того, отсортирован ли массив или находится в случайном порядке.

5. Адаптивность к изменениям в данных: Сортировка кучей может быть эффективной в обработке динамических данных или при небольших изменениях в массиве данных. Однако, в общем случае, если вставляются или удаляются элементы, то перестройка кучи может потребоваться снова для поддержания структуры.

6. Не требует дополнительной памяти: Сортировка кучей не требует дополнительной памяти, за исключением небольшого объема памяти, используемого для временных переменных при реализации алгоритма.



This work is licensed under
a Creative Commons Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of
Kelly Loves Whales and Nick Merritt.