**Heapsort**

aka Пирамидальная сортировка

aka Сортировка кучей

**Историческая справка**

Была изобретена Джоном Уильямсом в 1964 году. 

Джон Уильямс

**Принцип работы**

Для создания бинарного дерева элемент с индексом i должен быть больше элементов с индексами 2i+1 и 2i+2.

Алгоритм создания бинарного дерева:

1. Для i элемента массива найти наибольший среди i, 2i+1, 2i+2;
2. Если наибольшим элементом не является i, то поменять наибольший и i местами и перейти к пункту 1 с i равным индексу наибольшего;
3. Повторить данный алгоритм при i = {length/2 – 1; 0}.

Алгоритм сортировки построенного бинарного дерева:

1. Поменять местами первый и последний элемент массива (первый всегда будет наибольшим);
2. Создать бинарное дерево под элементом с индексом i = 0
3. Повторить данный алгоритм с массивом длиной n-1 (без последнего наибольшего элемента).

**Временная сложность**

* Временная сложность создания бинарного дерева под i элементом O(log n), потому что повторный вызов происходит при индексе, умноженном на 2.
* Временная сложность создания бинарного дерева из всего массива O(n). Не смотря на то, что мы проводим O(log n) n/2 раз, можно заметить, что при проходе массива некоторые ветви дерева уже построены, что и даёт нам сложность O(n).
* При сортировке дерева мы n раз проводим O(log n) операцию, что даёт нам сложность O(n log n).

**Графики**

Графики временной сложности:

Array (на первом графике он слился с осью OX)

LinkedList

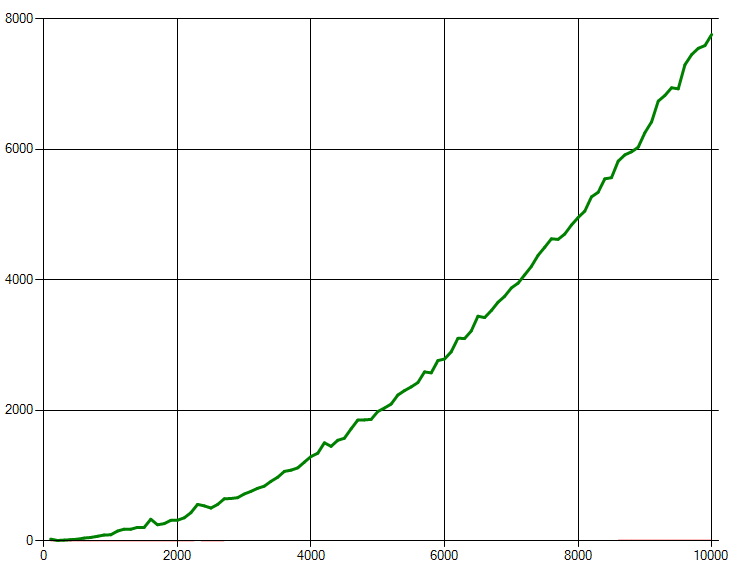
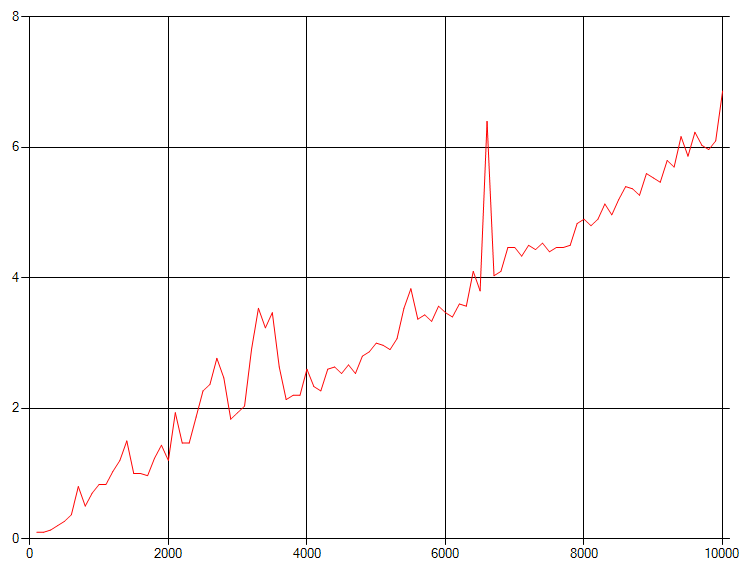


График без LinkedList для показания работы Array



Графики итераций:

Array (на первом графике он слился с осью OX)

LinkedList

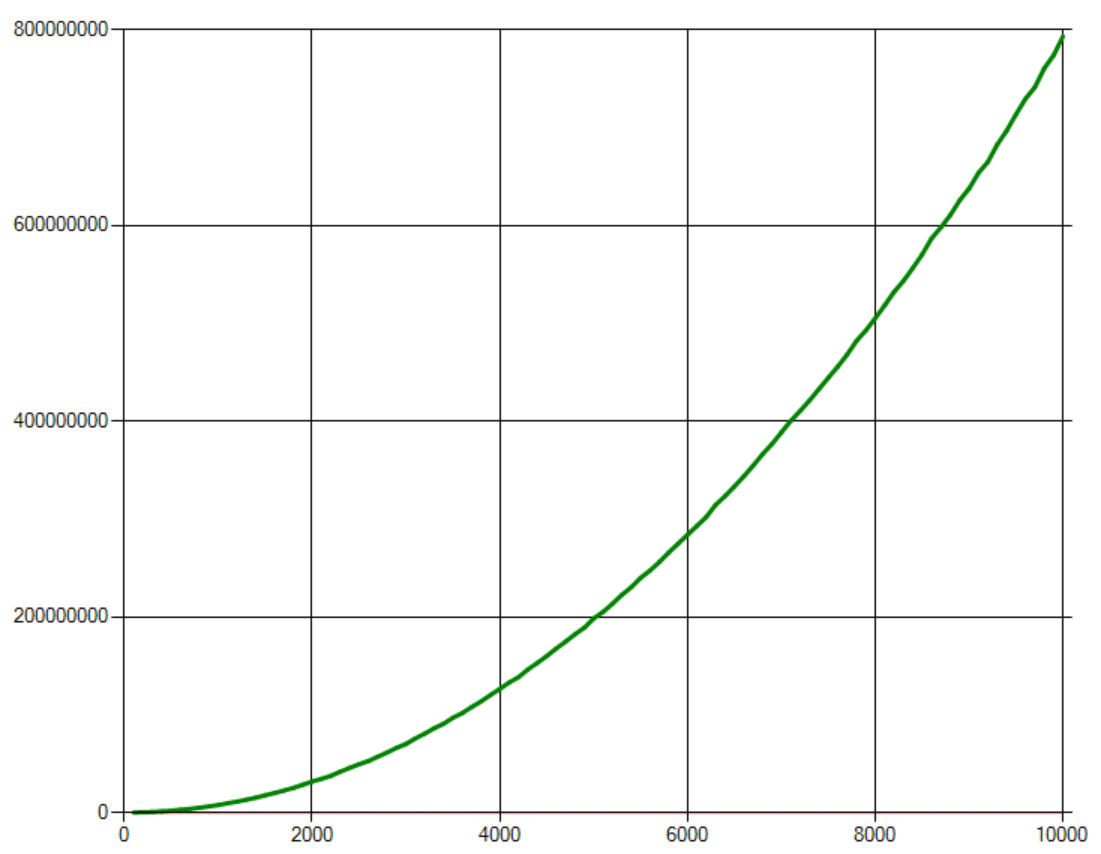
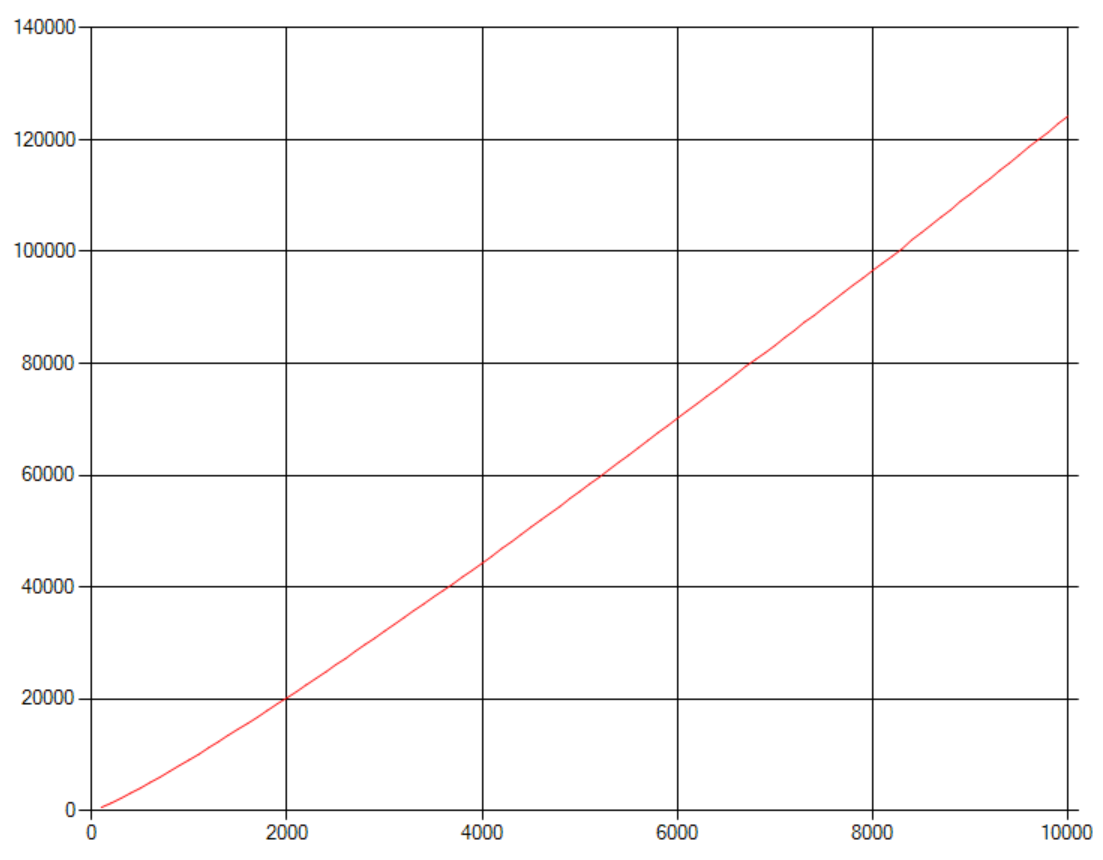


График без LinkedList для показания работы Array



**Выводы**

Плюсы:

* Доказанная сложность O(n log n)
* Не требует дополнительной памяти

Минусы:

* Неустойчив
* На почти отсортированных массивах работает столько же, сколько и на хаотичных данных
* Требует мгновенный доступ к памяти

Итого:

Сортировку следует использовать на малом количестве длинных массивов, т.к. для каждого массива требуется подготовка в виде создания из него бинарного дерева, но на длинных массивах он работает так же эффективно, как и на коротких.

**Использованная литература**

wikipedia.org

geeksforgeeks.org

Код реализации можно найти на моём git репозитории:

https://github.com/RoyalJackal/Orlov\_Dmitry\_11-808/tree/master/2019/winter/aisd/heapsort