TREE SORT

Сортировка двоичным деревом

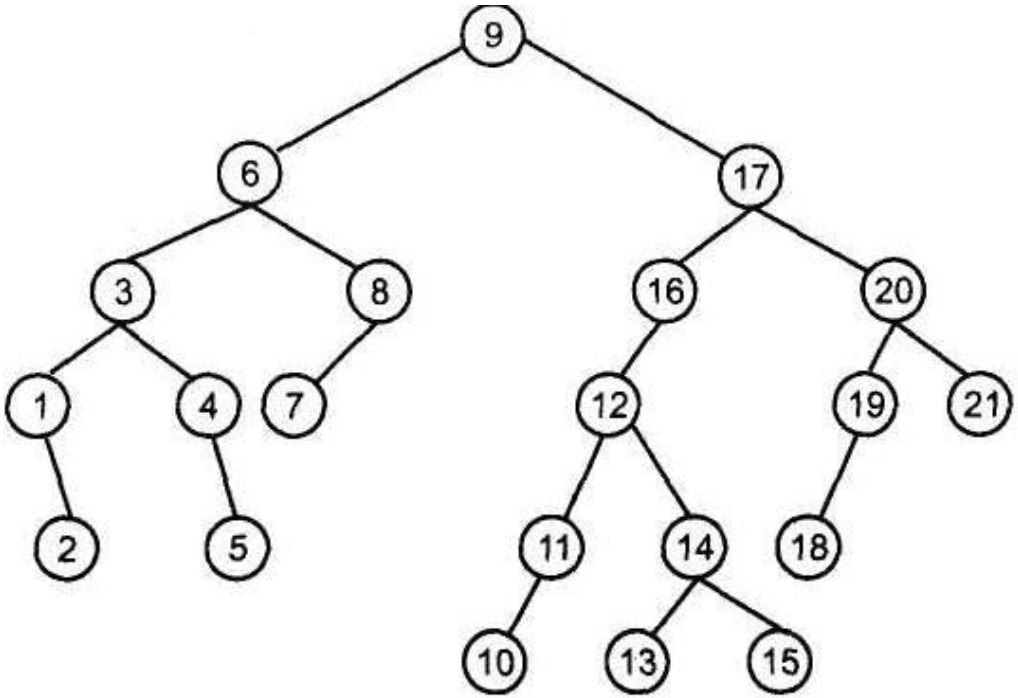
Что это?

Сортировка с помощью двоичного дерева — универсальный алгоритм сортировки, заключающийся в построении двоичного дерева поиска по ключам массива (списка), с последующей сборкой результирующего массива путём обхода узлов построенного дерева в необходимом порядке следования ключей.

Алгоритм

1. Построение двоичного дерева.

2. Сборка результирующего массива путём обхода узлов в необходимом порядке следования ключей.



В чем недостаток алгоритма?

При физическом развёртывании древовидной структуры в памяти требуется не менее чем 4n ячеек дополнительной памяти.

Каждый узел содержит:

- ссылки на элемент исходного массива;
- на родительский элемент;
- на левый и правый лист.

Однако, существуют способы уменьшить требуемую дополнительную память.

Эффективность

Процедура добавления объекта в бинарное дерево имеет среднюю алгоритмическую сложность порядка O(log(n)).

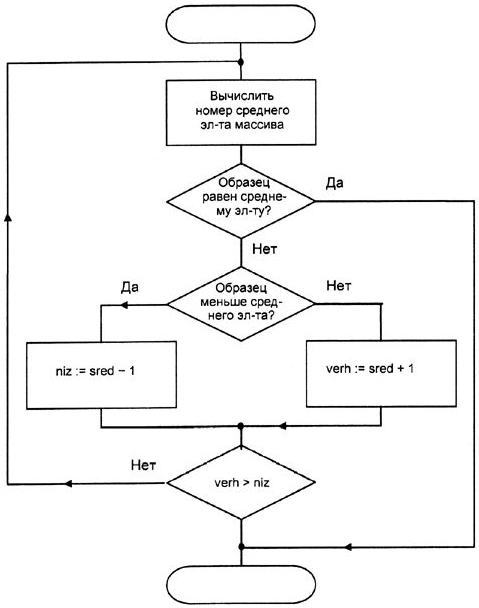
Однако, сложность добавления объекта в разбалансированное дерево может достигать O(n), что может привести к общей сложности порядка $O(n^2)$!!!

В чем причина роста сложности алгоритма?

Общее быстродействие метода O(nlogn). Почему?

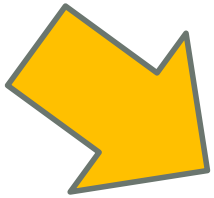
Поведение неестественно, устойчивости, вообще говоря, нет.

Требуется n операций на создание первоначального дерева, каждый по log n сравнений для выбора наименьшего и наибольшего.



Результаты работы алгоритма

Размер массива	Количество итераций	Время (ннс)
100	5049	1805108
200	20099	981807
300	45149	661719
400	80199	838690
500	125249	1292147
600	180299	581697
700	245349	592982
800	320399	779700
900	405449	777648
1000	500499	908965
1100	605549	1116202
1200	720599	981293
1300	845649	1467067
1400	980699	1327029
1500	1125749	1579918
1600	1280799	1881539
1700	1445849	2100060
1800	1620899	1762020
1900	1805949	1818958
2000	2000999	1482969
2100	2206049	1299329
2200	2421099	1920011
2300	2646149	1694309
2400	2881199	1580944
2500	3126249	3675362
2600	3381299	1885643
2700	3646349	3566614
2800	3921399	2737157
2900	4206449	2491962
3000	4501499	2380650



3100	4806549	2873605
3200	5121599	3276278
3300	5446649	2746391
3400	5781699	3143934
3500	6126749	3643558
3600	6481799	3844639
3700	6846849	3674849
3800	7221899	4085730
3900	7606949	3987242
4000	8001999	4979307
4100	8407049	4787460
4200	8822099	5425070
4300	9247149	5440458
4400	9682199	5447127
4500	10127249	5804148
4600	10582299	6572050
4700	11047349	6969594
4800	11522399	7145027
4900	12007449	6494079
5000	12502499	6883417

График зависимости затраченного времени от размера массива.

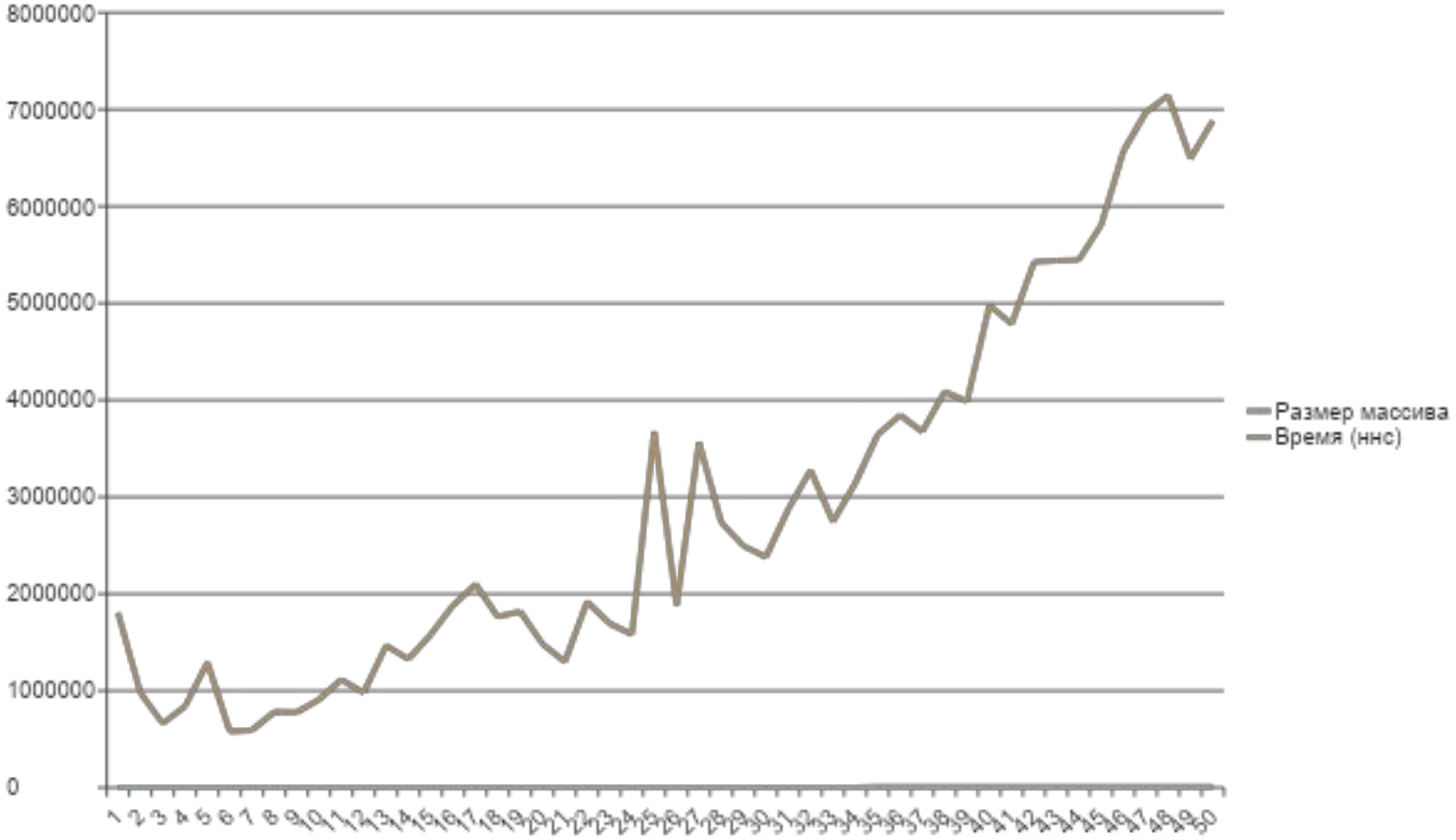
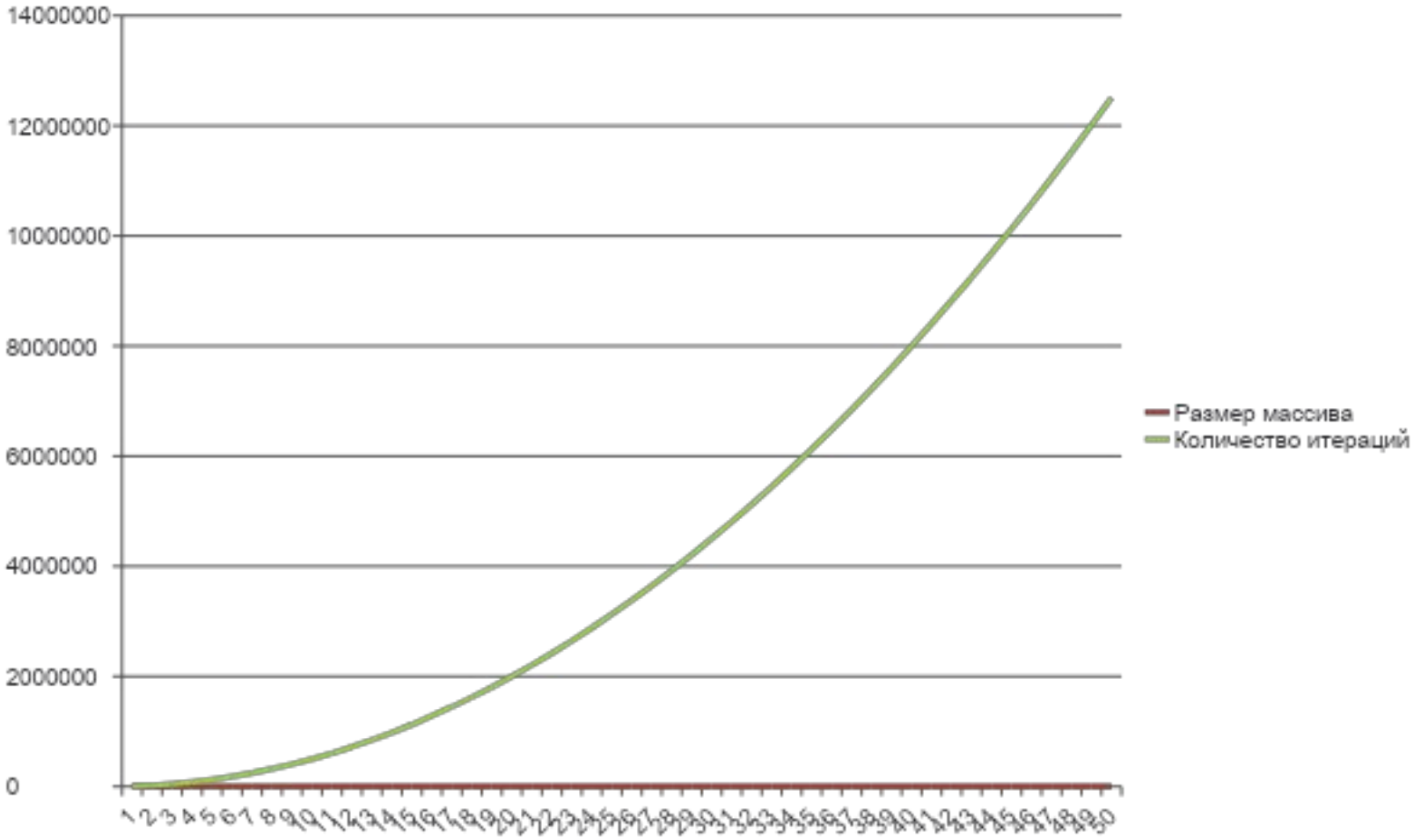


График зависимости количества итераций от размера массива.



Применение

TreeSort обычно применяют там, где –

- построенное дерево можно с успехом применить для таких задач;
- данные уже построены в 'дерево‘;
- данные можно считывать непосредственно в дерево;
- при потоковом вводе с консоли или из файла.

Источники

- http://cppalgo.blogspot.ru/2010/09/treesort.html
- https://ru.wikipedia.org
- http://www.martinbroadhurst.com/cpp-sorting.html
- Описание метода 'древесной сортровки' можно найти у Н. Вирта в книге 'Алгоритмы + Структуры Данных = Программы'