УДК 004.021

АНАЛИЗ АЛГОРИТМОВ СОРТИРОВКИ СО СРЕДНЕЙ АССИМПТОТИКОЙ NLOGN

А.А. Файтельсон
Бакалавр первого года обучения по направлению подготовки «Математическое обеспечение и администрирование информационных систем»
Курский государственный университет
e-mail: z0tedd@gmail.com

Научный руководитель:
В.А. Кудинов
Доктор педагогических наук, профессор, профессор кафедры программного обеспечения и администрирования информационных систем.
Курский государственный университет

Статья посвящена рассмотрению различных алгоритмов сортировки, а также проблеме выбора, подходящего из них. Эффективность каждого алгоритма экспериментальна подтверждена с помощью уже разработанных программ для отслеживания процессорного времени выполнения.

Ключевые слова: *алгоритм сортировки, сравнение алгоритмов сортировки.*

Введение. В современном мире важностью алгоритмов обработки данных нельзя недооценивать. Данная область требует постоянного совершенствования методов анализа взаимодействия с информацией. С растущим каждую секунду объемом цифровых объектов задача выбора необходимого алгоритма становится все более актуальной. И именно по этой причине в данной работе будут рассмотрены алгоритмы со средним асимптотическим временем nlogn.

Прежде, чем углубиться в сами алгоритмы, стоит упомянуть, что такое асимптотическое время. Асимптотическое время — это оценка временной сложности алгоритма, которая указывает на поведение алгоритма при стремлении размера входных данных к бесконечности. Это позволяет оценить, насколько быстро будет работать алгоритм при увеличении объема данных. В научных статьях асимптотическое время часто используется для сравнения эффективности различных алгоритмов и определения их временной сложности. Временной сложностью обычно называют количество элементарных операций, совершенными алгоритмом. Время одной такой операции есть некоторая константа, в нотации «O» большое, время оценивается как O(1). В данной нотации учитывается только слагаемое самого высокого порядка, поэтому в независимости от самого значения констан-

ты, она считается, как 1. Время работы алгоритма отличается от самих входных данных, поэтому чаще всего используется время работы в худшем и среднем случае.

Алгоритмы сортировки со среднем временем O(nlogn) опираются либо на рекурсивный метод «разделяй и властвуй», либо на нестандартные структуры данных в виде кучи или двоичного дерева поиска. Перед рассмотрением алгоритмов необходимо разобраться с этими понятиями.

Метод «Разделяй и властвуй» заключается в разделении одной большой задачи на маленькие ее части, нахождения решения для каждой подзадачи и в конце объединение полученных результатов в финальное решение изначальной задачи [1]. Фактически сам метод можно представить в виде двух шагов [2]:

1) Определение простейшего случая(базового)
2) Дробление исходной задачи до тех пор, пока она не будет сведена до базового случая

На основе данного метода основаны алгоритмы быстрой сортировки (quick sort) и сортировки слияния (merge sort).

Другие две сортировки, рассматриваемые в этой статье, опираются на определенную структуру данных – деревья. Один из алгоритмов основан на вставку элементов в двоичное дерево поиска, а другой на представлении исходного массива в виде кучи.

Дерево – структура, в которой у каждого узла(элементарной единицы структуры) есть 0 или более подузлом(«потомков») [3].

Двоичное дерево поиска – дерево, для которого выполняется ряд условий ([4] рис.1):

1. Каждый узел имеет не более двух потомков.
2. Любое поддерево любого узла является двоичным деревом.
3. Значения узла больше или равно значению подузла левого поддерева.
4. Значения узла меньше или равно значению подузла правого поддерева.

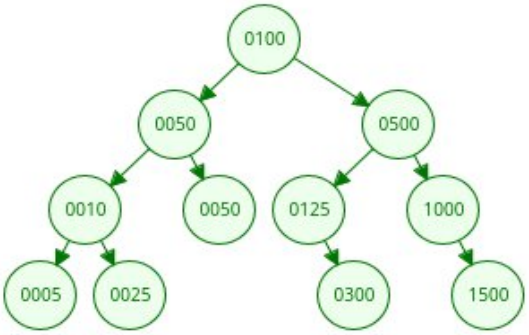

Рисунок 1. - Пример двоичного дерева поиска.

Куча – дерево ([5] рис.2), в котором значение любого узла больше или равно значению у его потомков. Конкретно в алгоритме пирамидальной сортировки используется двоичная куча (сортирующее дерево), поэтому к основному условию добавляются еще два:

1. Каждый узел имеет не более двух потомков.
2. Слои заполняются последовательно сверху вниз и слева направо, без пробелов [6].

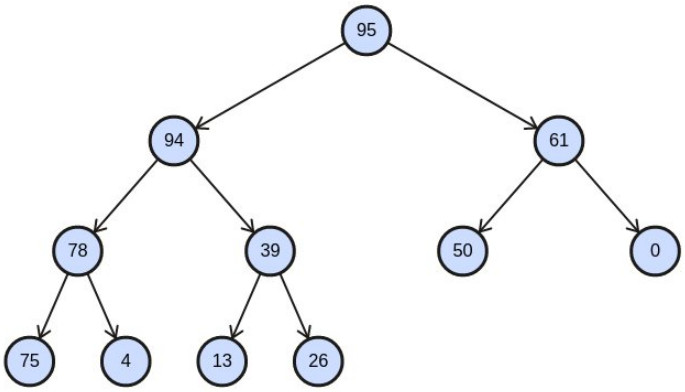

Рисунок 2. – Пример двоичной кучи.

Данные структуры данных крайне эффективны из-за своих свойств, что и делает алгоритмы сортировки, основанные на них крайне быстрыми.

Быстрая сортировка. Алгоритм начинается с выбора опорного элемент, относительно которого массив делится на две части: больше опорного и меньше его. Эффективное деление массива реализует схема Хоара, она использует два указателя, которые идут с противоположных частей структуры. Данные указатели двигаются в сторону друг друга, пока не встретится пара элементов, где один из них меньше опорного и располагается после него, а второй располагается после и при этом больше опорного. После разделения массива на две половинки, алгоритм применяется заново для каждой из частей, пока мы не получим полностью отсортированный массив. ([7] рис.3)

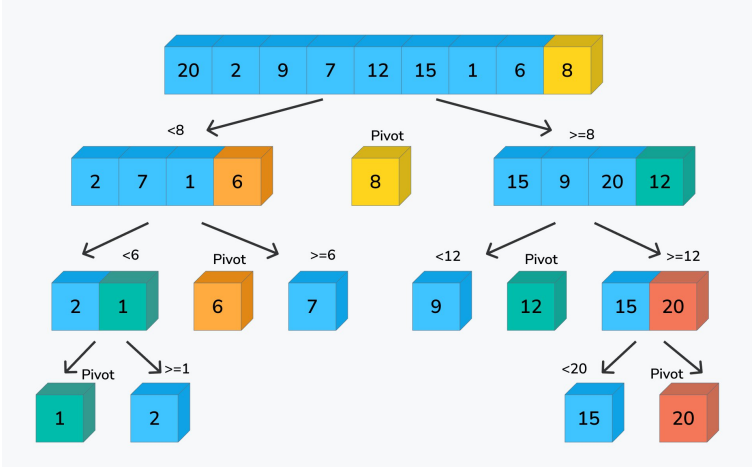

Рисунок 3. – Визуализация алгоритма быстрой сортировки.

Характеристика алгоритма:

1. Асимптотическая оценка времени
   - ♦ Время в худшем случае – O(n^2)
   - ♦ Время в лучшем случае – O(nlogn)
   - ♦ Время в среднем случае – O(nlogn)
2. Оценка потребления памяти
   - ♦ Память в худшем случае – O(n)
   - ♦ Память в лучшем случае – O(logn)
   - ♦ Память в среднем случае – O(logn)
3. Реальная оценка времени

Таблица 1 – время работы алгоритма в сек. на переменных данных

№/N	1.00E+01	1.00E+02	1.00E+03	1.00E+04	1.00E+05	1.00E+06	1.00E+07	1.00E+08	1.00E+09
1	0.006	0.037	0.447	0.941	11.298	137.715	1559.13	18361.5	199432
2	0.001	0.006	0.079	0.954	11.541	135.532	1568.48	17110.3	201431
3	0.001	0.007	0.108	1.068	12.078	141.431	1595.41	17875.3	199493
4	0.002	0.007	0.082	1.032	11.945	146.68	1698.7	18425.5	197540
5	0.001	0.006	0.078	0.976	11.528	136.259	1564.94	17812.9	196614
6	0.001	0.006	0.078	0.96	11.493	134.68	1565.83	17482.2	196442
7	0.001	0.006	0.079	0.964	11.572	135.219	1559.06	17553.5	196444
8	0.001	0.006	0.078	0.956	11.53	135.363	1551.23	17582.1	195296
9	0.001	0.007	0.079	0.946	11.541	136.246	1545.84	17513.1	195353
10	0.001	0.006	0.077	0.992	11.512	134.369	1561.52	18036.3	197161
Ср.знач.	0.0016	0.0094	0.1185	0.9789	11.6038	137.3494	1577.014	17775.27	197520.6

Сортировка слиянием. Данный алгоритм также основан на парадигме «разделяй и властвуй», как и быстрая сортировка. Массив рекурсивно делится на две части, которые также делятся на две части, пока мы не придем к массивам длиной 1 или 2. Далее выполняется слияние полученных массивов в другой. Берутся два указателя на начала двух подмассивов, из этих указателей берется с минимальным элементом, и вставляется в новый массив, а указатель сдвигается вправо. Операция происходит до того момента, пока все подмассивы не будут слиты воедино.([7] рис. 4) Само слияние выполняется за O(n), но так как мы имеем O(logn) подмассивов, то сама сортировка выполняется за O(nlogn).

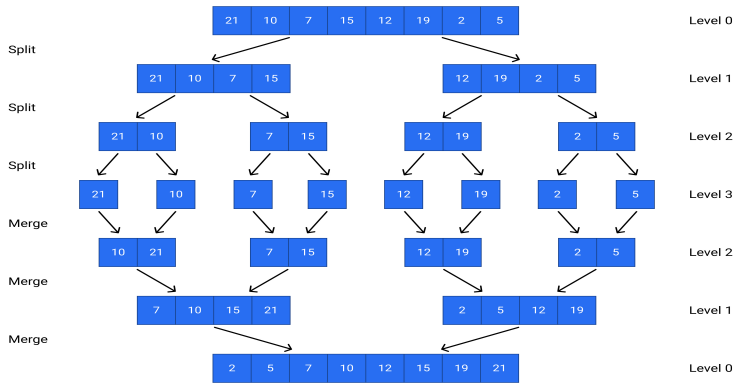

Рисунок 4. – Визуализация алгоритма сортировки слиянием.

Характеристика алгоритма:

1. Асимптотическая оценка времени
   - Время в худшем случае – O(nlogn)
   - Время в лучшем случае – O(nlogn)
   - Время в среднем случае – O(nlogn)
2. Оценка потребления памяти
   - Память в худшем случае – O(n)
   - Память в лучшем случае – O(n)
   - Память в среднем случае – O(n)
3. Реальная оценка времени

Таблица 2 – время работы алгоритма в сек. на переменных данных

№/N	1,00E+01	1,00E+02	1,00E+03	1,00E+04	1,00E+05	1,00E+06	1,00E+07	1,00E+08	1,00E+09
1	0,01	0,046	0,455	1	12,275	149,883	1672,21	19038,7	213812
2	0,002	0,007	0,081	0,961	12,044	144,019	1681,83	18962,4	213724
3	0,002	0,007	0,079	0,982	12,167	143,912	1676,86	18916,3	213207
4	0,002	0,008	0,081	1,001	12,592	147,222	1674,56	18973,2	213316
5	0,002	0,007	0,078	0,986	12,099	144,247	1677,03	18920,5	213771
6	0,002	0,007	0,091	1,022	12,308	143,701	1674,03	18972,8	213261
7	0,002	0,007	0,08	0,984	12,263	143,945	1674,61	19042,8	213389
8	0,001	0,006	0,083	0,982	12,341	147,897	1672,18	18935,1	212485
9	0,002	0,006	0,091	0,962	12,07	148,132	1673,15	18872,7	213458
10	0,002	0,007	0,081	1,29	12,659	153,688	1674,37	18933,3	213703
Ср.знач.	0,0027	0,0108	0,12	1,017	12,2818	146,665	1675,08	18956,8	213413

Сортировка кучей. В начале алгоритма из массива за O(n) строится куча.

Так как в корне кучи лежит максимальный(минимальный) элемент, то извлекая его мы можем построить новый отсортированный массив. При извлечении элемента мы «портим» кучу, поэтому, чтобы вернуть свойства данной структуры мы должны ее «просеить», делается это за O(logn). Последовательно извлекая и просеивая кучу из n элементов, получаем время O(nlogn). ([8] рис. 5)

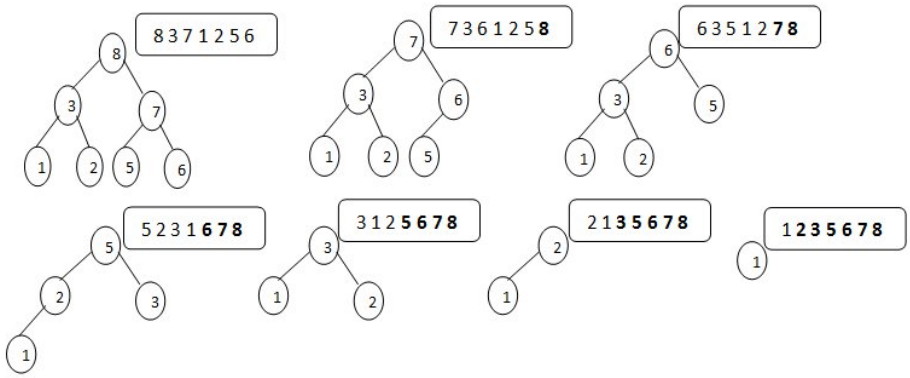

Рисунок 5. – Визуализация алгоритма сортировки кучей.

Характеристика алгоритма:

1. Асимптотическая оценка времени
   - ♦ Время в худшем случае – O(nlogn)
   - ♦ Время в лучшем случае – O(nlogn)
   - ♦ Время в среднем случае – O(nlogn)
2. Оценка потребления памяти
   - ♦ Память в худшем случае – O(1)
   - ♦ Память в лучшем случае – O(1)
   - ♦ Память в среднем случае – O(1)
3. Реальная оценка времени

Таблица 3 – время работы алгоритма в сек. на переменных данных

№/N	1,00E+01	1,00E+02	1,00E+03	1,00E+04	1,00E+05	1,00E+06	1,00E+07	1,00E+08	1,00E+09
1	0,003	0,019	0,321	3,056	16,126	202,351	2799,64	43851,1	599768
2	0,001	0,007	0,093	1,197	15,242	194,773	2906,18	44298,1	578503
3	0,001	0,007	0,092	1,202	15,275	195,548	2968,22	44343,4	535452
4	0,001	0,007	0,094	1,197	15,272	195,44	2847,31	41460,8	593033
5	0,001	0,007	0,095	1,222	15,78	201,691	2962,02	45040,5	601154
6	0,005	0,008	0,095	1,286	17,307	269,238	4454,41	51166,8	574428
7	0,002	0,007	0,096	1,318	15,778	204,846	3034,18	44882,1	581353
8	0,001	0,007	0,093	1,207	15,452	222,828	3021,38	45795,9	595226
9	0,002	0,007	0,095	1,221	15,669	206,269	3153,08	44756,9	591862
10	0,001	0,007	0,113	1,224	15,704	195,427	3076,22	44333,9	617594
Ср.знач.	0,0018	0,0083	0,1187	1,413	15,761	208,84	3122,3	44993	586837

Сортировка двоичным деревом поиска. Данный алгоритм строит двоичное дерево поиска из элементов массива, а после этого обходит эту структуру и строит результирующий массив. (рис. 6)

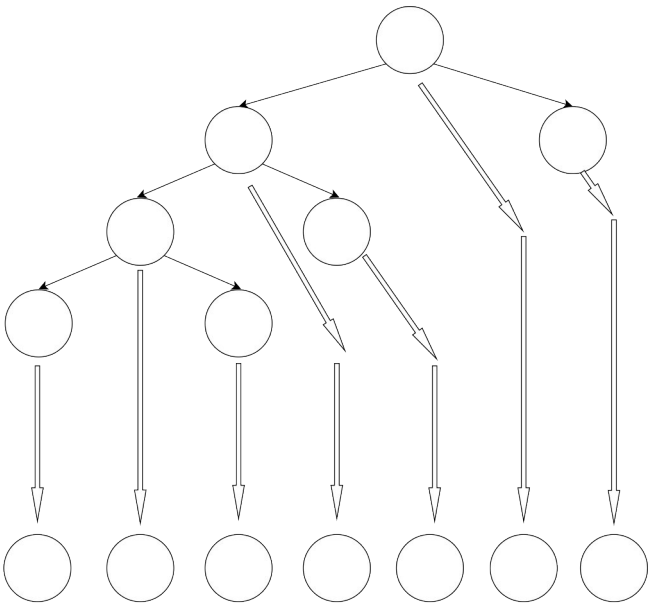

Рисунок 6. – Визуализация алгоритма сортировки деревом

Характеристика алгоритма:

1. Асимптотическая оценка времени
   - Время в худшем случае – O(n^2)
   - Время в лучшем случае – O(nlogn)
   - Время в среднем случае – O(nlogn)
2. Оценка потребления памяти
   - Память в худшем случае – O(n)
   - Память в лучшем случае – O(n)
   - Память в среднем случае – O(n)
3. Реальная оценка времени

Таблица 4 – время работы алгоритма в сек. на переменных данных

№/N	1,00E+01	1,00E+02	1,00E+03	1,00E+04	1,00E+05	1,00E+06	1,00E+07	1,00E+08
1	0,008	0,052	0,646	10,128	45,754	553,253	10727,20	173761,0
2	0,003	0,009	0,128	2,081	68,279	1206,530	14358,20	185458,0
3	0,008	0,051	0,590	1,763	30,912	482,927	9428,32	148620,0
4	0,002	0,012	0,138	1,814	31,523	481,91	9250,34	148929,0
5	0,001	0,012	0,130	1,759	28,34	480,136	9295,58	148309,0
6	0,006	0,008	0,126	1,746	30,819	474,900	9180,39	148683,0
7	0,002	0,009	0,132	1,695	29,491	477,117	9154,59	150574,0
8	0,002	0,012	0,111	1,683	27,505	477,566	9191,88	150190,0
9	0,001	0,011	0,119	1,787	29,031	548,129	9279,37	150914,0
10	0,009	0,044	0,661	9,174	34,256	478,838	9213,36	149241,0
Ср.знач.	0,0042	0,022	0,2781	3,363	35,591	566,13	9907,9	155468

Заключение. Из представленных таблиц можно сделать вывод, что не все сортировки с ассимптотически одинаковым временем имеют одинаковую производительности. Все сортировки, кроме сортировки двоичным деревом поиска эффективно показывают себя на больших размерах массива. На основе полученных данных составим 2 рейтинга алгоритмов по скорости выполнения на массиве из 100.000.000 чисел и по использованию памяти.

Рейтинг из рассмотренных алгоритмов по скорости:

1. Быстрая сортировка
2. Сортировка слиянием
3. Сортировка кучей
4. Сортировка двоичным деревом поиска

Рейтинг из рассмотренных алгоритмов по памяти:

1. Сортировка кучей
2. Быстрая сортировка
3. Сортировка слиянием, сортировка двоичным деревом поиска

Таким образом, можно сделать о вывод о необходимости выбора правильного алгоритма сортировки в зависимости от ограничений к разрабатываемой программы.

Список используемой литературы

1. *Основы алгоритмов* Яндекс образование [сайт]. Официальный сайт. 2024 год URL:https://education.yandex.ru/handbook/algorithms/article/razdelyaj-i-vlastvuj (дата обращения: 12.03.2024).
2. *Адитья Бхаргава* Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих: пер. с англ. / изд. «Питер», 2015. – 88 с.
3. *TProger* Алгоритмы и структуры данных для начинающих: двоичное дерево поиска. Официальный сайт. 2024 год URL:https://tproger.ru/translations/binary-search-tree-for-beginners(дата обращения: 12.03.2024).
4. *University of San Francisco* Data Structure Visualizations Официальный сайт. 2024 год URL:https://www.cs.usfca.edu/~galles/visualization/BST.html (дата обращения: 12.03.2024).
5. BinaryTreeVisualiser Официальный сайт. 2024 год URL: http://btv.melezinek.cz/binary-heap.html(дата обращения: 12.03.2024).
6. Algorithmica Официальный сайт. 2024 год URL: https://ru.algorithmica.org/cs/basic-structures/heap/(дата обращения: 12.03.2024).
7. WorkatTech Официальный сайт. 2024 год URL: https://workat.tech/problem-solving/tutorial/sorting-algorithms-quick-sort-merge-sort-dsa-tutorials-6j3h98lk6j2w(дата обращения: 12.03.2024).
8. Essential Algorithms Essential Programming Books URL:https://www.programming-books.io/essential/algorithms/heap-sort-basic-information-3193e2927dbe4c03bcbc5645fa66cf21 (дата обращения: 12.03.2024).