

СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ АЛГОРИТМОВ ПОИСКА И СОРТИРОВОК НА МАССИВАХ

Малюш Д.О., студент гр. 250505, Борисенко К.Н., студент гр. 250501

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Сидорович А.С. – старший преподаватель

В научной работе обсуждается выбор оптимальных алгоритмов поиска и сортировки данных в массивах. Это предполагает сравнительный анализ алгоритмов.

Существует множество способов поиска необходимой позиции в массиве и алгоритмов сортировки массива. Для исследования сравнительных характеристик алгоритмов поиска и сортировок будет рассматриваться массив, состоящий из 100 символов.

Алгоритм поиска в массиве

Линейный поиск. Линейный поиск — это простой алгоритм поиска, который работает путем последовательного просмотра каждого элемента в массиве и сравнения его с искомым значением. Если элемент найден, алгоритм возвращает его индекс[5].

Индексно-последовательный поиск. Индексно-последовательный поиск — это алгоритм поиска, который использует индекс для поиска элемента в списке или массиве. Затем алгоритм использует индекс для быстрого нахождения диапазона элементов, которые могут содержать искомое значение. После этого алгоритм выполняет последовательный поиск в этом диапазоне для нахождения искомого элемента. Массив изначально должен быть отсортирован.

Бинарный (двоичный) поиск. Бинарный поиск также известен как логарифмический поиск. Он работает путем деления массива на две части и сравнения искомого значения с элементом в середине[1]. Если искомое значение меньше значения в середине массива, то поиск продолжается в левой половине массива. В другом случае поиск продолжается в правой половине массива. Этот процесс повторяется до тех пор, пока не будет найден искомый элемент или пока не останется ни одного элемента для поиска.

Поиск прыжками. Поиск прыжками - это алгоритм поиска, требует предварительной сортировки набора данных[2].

Алгоритм работает путем прыжков через фиксированное количество элементов в массиве и сравнения каждого прыжка с элементом, который мы должны найти. Если текущий прыжок больше искомого элемента, то поиск продолжается в предыдущем интервале прыжка с помощью линейного поиска.

Сравнение скорости работы перечисленных алгоритмов поиска представлено на рисунке 1.

Сравнение памяти, необходимой для работы перечисленных алгоритмов поиска представлено на рисунке 2.

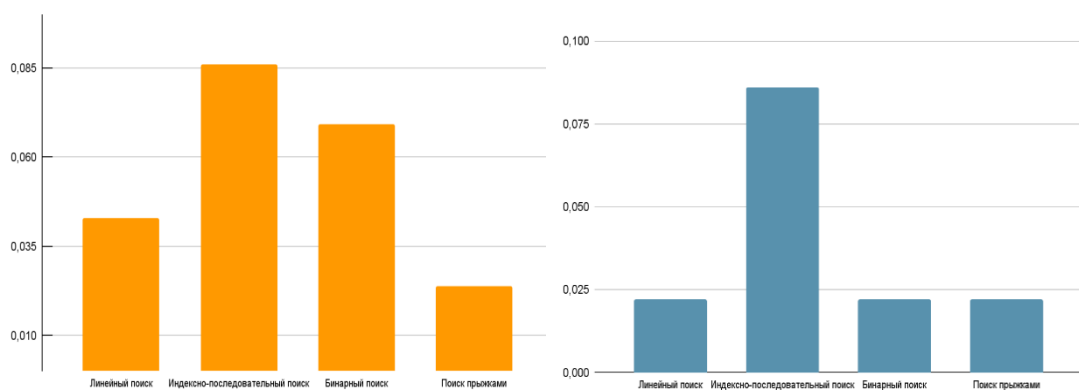


Рисунок 1 – Диаграмма, отображающая различие в скорости работы рассматриваемых алгоритмов поиска

Рисунок 2 – Диаграмма, отображающая различие в количестве памяти, требуемой для работы рассматриваемых алгоритмов поиска

Сортировка массива

Сортировка пузырьком. Сортировка пузырьком — это простой алгоритм сортировки. Он работает путем многократного сравнения и обмена соседних элементов, если они находятся в неправильном порядке[3]. Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется перестановка элементов.

Быстрая сортировка (сортировка Хоара): Быстрая сортировка — это алгоритм сортировки. Состоит из трех шагов: выбор опорного элемента из массива, разбиение массива таким образом, чтобы элементы меньше опорного помещались перед ним, а большие или равные - после, и рекурсивное повторение этих действий для левой и правой части массива[3].

Сортировка методом вставок. Сортировка вставками — это алгоритм сортировки, который работает путем перемещения текущего элемента входной последовательности на нужную позицию в уже отсортированном массиве. Отсортированный массив строится по одному элементу за раз[4]. Внутри функции используется цикл, который начинается с первого значения массива и продолжается до конца массива. На каждой итерации цикла выбирается новый элемент из массива и сравнивается с элементами в уже отсортированном списке. Если порядок между сравниваемыми элементами нарушен, то меняем их местами.

Сортировка методом Шелла. Сортировка Шелла — алгоритм сортировки, являющийся усовершенствованным вариантом сортировки вставками. Идея метода Шелла состоит в сравнении элементов, стоящих не только рядом, но и на определенном расстоянии друг от друга[3].

При сортировке Шелла сначала сравниваются и сортируются между собой значения, стоящие один от другого на некотором расстоянии. После этого процедура повторяется для некоторых меньших значений.

Сравнение скорости работы перечисленных алгоритмов сортировки представлено на рисунке 3.

Сравнение памяти, необходимой для работы перечисленных алгоритмов сортировки представлено на рисунке 4.

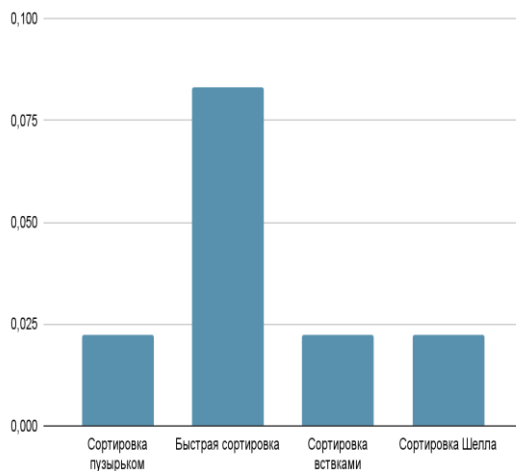


Рисунок 3 – Диаграмма, отображающая различие в скорости работы рассматриваемых алгоритмов сортировок.

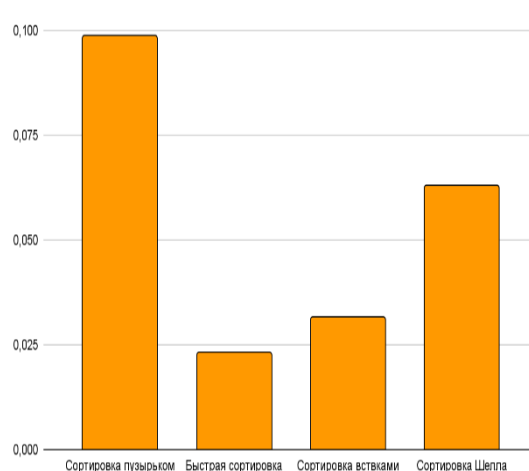


Рисунок 4 – Диаграмма, отображающая различие в количестве памяти, требуемой для работы рассматриваемых алгоритмов сортировки

Каждый способ сортировки и алгоритм поиска будет уместен в той или иной ситуации. Для максимально быстрой сортировки лучше использовать сортировку Хоара, ведь этот алгоритм имеет наибольшую скорость, а самым эффективным алгоритмом поиска по всем параметрам оказался именно алгоритм поиска прыжками. Зная данную информацию, можно выбирать наиболее подходящие алгоритмы поиска или сортировки, для того чтобы оптимизировать и ускорить работу программы.

Список использованных источников:

1. Левитин А. В. Глава 4. Метод декомпозиции: Бинарный поиск // *Алгоритмы. Введение в разработку и анализ* — М.: Вильямс, 2006. — С. 180—183. — 576 с.
2. Скиена С. *Алгоритмы. Руководство по разработке*. 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург. 2011. — 720 с.: ил.
3. Липачёв Е. К. // *Технология программирования. Методы сортировки данных // учебное пособие / Е.К. Липачёв. — Казань: Казан. ун-т, 2017. — 58 с.*
4. Кнут Д. Э. 5.2 Внутренняя сортировка // *Искусство программирования. Том 3. Сортировка и поиск = The Art of Computer Programming. Volume 3. Sorting and Searching / под ред. В. Т. Тертышного (гл. 5) и И. В. Красикова (гл. 6). — 2-е изд. — Москва: Вильямс, 2007. — Т. 3. — 832 с..*
5. Кнут Д. *Искусство программирования, том 3. Сортировка и поиск = The Art of Computer Programming, vol.3. Sorting and Searching. — 2-е изд. — М.: «Вильямс», 2007. — С. 824*