Московский авиационный институт   
(государственный технический университет)   
  
Факультет прикладной математики   
  
Кафедра вычислительной математики и программирования

Курсовой проект по курсам

“Языки и методы программирования” и “Алгоритмы и структуры данных”

8 факультет, 2 курс, весенний семестр 2020/21 учебного года

Студент: Стрыгин Д.Д.

Группа: М8О-106Б-19, №20

Преподаватель: Дубинин А.В.

Содержание:

* Введение
* Задание
* Таблицы
* Сортировки
* Шейкерная сортировка
* Бинарный поиск
* Описание программы
* Заключение
* Список источников

Введение

Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методами двоичного поиска по ключу в таблице.

Программа должна вводить значения элементов неупорядоченной таблицы и проверять работу процедуры сортировки в трёх случаях: (1) элементы таблицы с самого начала упорядочены; (2) элементы таблицы расставлены в обратном порядке; (3) элементы таблицы не упорядочены. В последнем случае можно использовать встроенные процедуры генерации псевдослучайных чисел.

Для каждого вызова процедуры сортировки необходимо печатать исходное состояние таблицы и результаты сортировки. После выполнения сортировки программа должна вводить ключи и для каждого из них выполнять поиск в упорядоченной таблице с помощью процедуры двоичного поиска и печатать найденные элементы, если они присутствуют в таблице.

В процессе отладки и тестирования рекомендуется использовать команды обработки текстовых файлов ОС ОМХ и переадресацию ввода-вывода. Тестовые данные необходимо заранее поместить в текстовые файлы.

В качестве текста для записей таблицы взять фрагмент стихотворения (группы 3-5), прозы (группы 1, 2) или изображение АЗСП-графики (группы 6-8). Каждый элемент таблицы, содержащий ключ и текст записи, распечатывать в отдельной строке.

Задание

Сортировка: Шейкер

Ключ: i8+d

Хранить: вместе

Таблицы

Таблица — способ структурирования данных. Представляет собой распределение данных по однотипным строкам и столбцам (графам). Таблицы широко используются в различных исследованиях и анализе данных. Таблицы также встречаются в СМИ, в рукописных материалах, в компьютерных программах и в дорожных знаках.

Таблица состоит из упорядоченного набора строк и столбцов:

* элементы строки сравниваются между собой и имеют смысловую связь
* в столбце перечислены свойства этих элементов, по которым они сравниваются
* обычно столбец имеет название
* его название может состоять из слова, фразы или числа
* пересечением строки и столбца является ячейка таблицы

Элементы таблицы могут быть по-разному сгруппированы или расположены. Кроме того, таблица может включать метаданные, аннотации, заголовок, колонтитул.

Сортировки

Алгоритм сортировки — это алгоритм для упорядочивания элементов в списке. В случае, когда элемент списка имеет несколько полей, поле, служащее критерием порядка, называется ключ сортировки. На практике в качестве ключа часто выступает число, а в остальных полях хранятся какие-либо данные, никак не влияющие на работу алгоритма.

Алгоритмы сортировки оцениваются по скорости выполнения и эффективности использования памяти:

**Время** — основной параметр, характеризующий быстродействие алгоритма. Называется также вычислительной сложностью. Для упорядочения важны худшее, среднее и лучшее поведение алгоритма в терминах мощности входного множества A. Если на вход алгоритму подаётся множество A, то обозначим n = |A|. Для типичного алгоритма хорошее поведение — это O(n log n) и плохое поведение — это O(n^2). Идеальное поведение для упорядочения — O(n). Алгоритмы сортировки, использующие только абстрактную операцию сравнения ключей, всегда нуждаются по меньшей мере в сравнениях. Тем не менее, существует алгоритм сортировки Хана с вычислительной сложностью O(n log log n log log log n), использующий тот факт, что пространство ключей ограничено (он чрезвычайно сложен, а за О-обозначением скрывается весьма большой коэффициент, что делает невозможным его применение в повседневной практике). Также существует понятие сортирующих сетей. Предполагая, что можно одновременно (например, при параллельном вычислении) проводить несколько сравнений, можно отсортировать n чисел за O(log^2 n) операций. При этом число n должно быть заранее известно;

**Память** — ряд алгоритмов требует выделения дополнительной памяти под временное хранение данных. Как правило, эти алгоритмы требуют O(log n) памяти. При оценке не учитывается место, которое занимает исходный массив и независящие от входной последовательности затраты, например, на хранение кода программы (так как всё это потребляет O(1)). Алгоритмы сортировки, не потребляющие дополнительной памяти, относят к сортировкам на месте.

Шейкерная сортировка

Сортировка перемешиванием, или Шейкерная сортировка, или двунаправленная ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Cocktail sort*) — разновидность [пузырьковой сортировки](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%BE%D0%BC). Анализируя метод пузырьковой сортировки, можно отметить два обстоятельства.

*Во-первых*, если при движении по части массива перестановки не происходят, то эта часть массива уже отсортирована и, следовательно, её можно исключить из рассмотрения.

*Во-вторых*, при движении от конца массива к началу минимальный элемент «всплывает» на первую позицию, а максимальный элемент сдвигается только на одну позицию вправо.

Эти две идеи приводят к следующим модификациям в методе пузырьковой сортировки. Границы рабочей части массива (то есть части массива, где происходит движение) устанавливаются в месте последнего обмена на каждой итерации. Массив просматривается поочередно справа налево и слева направо.

Следующая таблица отражает временную сложность алгоритма шейкерной сортировки для трех случаев.

|  |  |  |
| --- | --- | --- |
| Лучший случай | Средний случай | Худший случай |
| O(n) | O(n2) | O(n2) |

Бинарный поиск

Бинарный поиск производится в упорядоченном массиве. При бинарном поиске искомый ключ сравнивается с ключом среднего элемента в массиве. Если они равны, то поиск успешен. В противном случае поиск осуществляется аналогично в левой или правой частях массива.

Алгоритм может быть определен в рекурсивной и нерекурсивной формах. Бинарный поиск также называют поиском методом деления отрезка пополам или дихотомии.  
Количество шагов поиска определится как

**log2n**,

где **n**-количество элементов

На каждом шаге осуществляется поиск середины отрезка по формуле

**mid = (left + right)/2**

Если искомый элемент равен элементу с индексом **mid**, поиск завершается.  
В случае если искомый элемент меньше элемента с индексом **mid**, на место **mid** перемещается правая граница рассматриваемого отрезка, в противном случае — левая граница.

Описание программы

Программа принимает на вход размер таблицы и саму таблицу. Далее идёт шейкерная сортировка таблицы и бинарный поиск элементов по ключу (алгоритмы сортировки и бинарного поиска описаны ранее).

Сложность поиска: O(log*n*).

Заключение

В курсовом проекте были рассмотрены сортировки и для чего они нужны. В ходе подготовки проекта была изучена сортировка - шейкер. В процессе выполнения проекта была написана программа, которая получает на вход таблицу и сортирует её, после чего находит элементы таблицы бинарным поиском по ключу.

Список источников:

* <https://ru.wikipedia.org/wiki/Таблица> Таблицы
* <https://prog-cpp.ru/search-binary/> Бинарный поиск
* <https://ru.wikipedia.org/wiki/Пирамидальная_сортировка> Пирамидальная сортировка
* <https://ru.wikipedia.org/wiki/Алгоритм_сортировки> Сортировки
* методичка