Индивидуальные задания для учебной практики
1 курс 1-ый семестр

Задание. Методы сортировки

Составить программу реализации указанного метода сортировки и иллюстрации его выполнения. В программе предусмотреть просмотр входных и выходных данных и пошаговое перемещение элементов в соответствии с алгоритмом.

Для получения входных данных иметь три варианта:

a) непосредственный ввод и запись в текстовый файл;
b) генерирование с помощью датчика случайных чисел и запись в текстовый файл;
c) ввод из текстового файла.

Алгоритм сортировки реализовать в виде процедуры с параметрами, которую поместить в отдельный файл.

Описание методов сортировки можно посмотреть в книгах:

1. *Лэнггсам Й., ОгенстайнМ., Тененбаум А. Структуры данных для персональных ЭВМ, М., Мир, 1989г., 567 стр.*

2. *Дональд Кнут* 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. — ISBN 5-8459-0082-4

3. *Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн* Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS. — 2-е изд. — М.: «Вильямс», 2006. — С. 1296. — ISBN 5-8459-0857-4

4. *Роберт Седжвик* Фундаментальные алгоритмы на C. Анализ/Структуры данных/Сортировка/Поиск = Algorithms in C. Fundamentals/Data Structures/Sorting/Searching. — СПб.: ДиаСофтЮП, 2003. — С. 672. — ISBN 5-93772-081-4

Список алгоритмов сортировки

Алгоритмы устойчивой сортировки

- Сортировка пузырьком (англ. *Bubble sort*) — сложность алгоритма: $O(n^2)$; для каждой пары индексов производится обмен, если элементы расположены не по порядку. **Дорофеев**
- Четно-нечетная сортировка **Заломов**
- Сортировка перемешиванием (Шейкерная, Cocktail sort, bidirectional bubble sort) — Сложность алгоритма: $O(n^2)$ **Стащенко**
- Гномья сортировка — имеет общее с сортировкой пузырьком и сортировкой вставками. Сложность алгоритма — $O(n^2)$.
- Сортировка вставками (метод простых ставок)(Insertion sort) — Сложность алгоритма: $O(n^2)$; определяем где текущий элемент должен находиться в упорядоченном списке и вставляем его туда **Атаев**
- Сортировка вставками (метод двухпутевых вставок)(Insertion sort) — Сложность алгоритма: $O(n^2)$; определяем где текущий элемент должен находиться в упорядоченном списке и вставляем его туда **Масюк**
- Блочная сортировка (Корзинная сортировка, черпачная сортировка, Bucket sort) — Сложность алгоритма: $O(n)$; требуется $O(k)$ дополнительной памяти и знание о природе сортируемых данных, выходящее за рамки функций "переставить" и "сравнить". **Степура**

- Сортировка подсчётом (Counting sort) — Сложность алгоритма: O(n+k); требуется O(n+k) дополнительной памяти (простой алгоритм) **Четвергов**
- Сортировка слиянием (Merge sort) — Сложность алгоритма: O(n log n); требуется O(n) дополнительной памяти; выстраиваем первую и вторую половину списка отдельно, а затем — сливаем упорядоченные списки (простое двухпутевое слияние) **Бочков**
- Сортировка слиянием (бинарное слияние)
- Сортировка слиянием (Алгоритм Пратта)
- Сортировка с помощью двоичного дерева (англ. *Tree sort*) — Сложность алгоритма: O(n log n); требуется O(n) дополнительной памяти **Шиляев**

Алгоритмы неустойчивой сортировки

- Сортировка выбором (Selection sort) — Сложность алгоритма: O(n^2); поиск наименьшего или наибольшего элемента и помещения его в начало или конец упорядоченного списка **Калинчук**
- Сортировка Шелла (Shell sort) — Сложность алгоритма: O(n $\log^2 n$); попытка улучшить сортировку вставками **Белкина**
- Сортировка расчёской (Comb sort) — Сложность алгоритма: O(n log n) **Сенькевич**
- Пирамидальная сортировка (Сортировка кучи, Heapsort) — Сложность алгоритма: O(n log n); превращаем список в кучу, берём наибольший элемент и добавляем его в конец списка
- Плавная сортировка (Smoothsort) — Сложность алгоритма: O(n log n)
- Быстрая сортировка (Quicksort) — Сложность алгоритма: O(n log n) — среднее время, O(n^2) — худший случай; широко известен как быстрейший из известных для упорядочения больших случайных списков; с разбиением исходного набора данных на две половины так, что любой элемент первой половины упорядочен относительно любого элемента второй половины; затем алгоритм применяется рекурсивно к каждой половине **Нестёркина**
-
- Introsort — Сложность алгоритма: O(n log n), сочетание быстрой и пирамидальной сортировки. Пирамидальная сортировка применяется в случае, если глубина рекурсии превышает log(n).
- Patience sorting — Сложность алгоритма: O(n log n + k) — наихудший случай, требует дополнительно O(n + k) памяти, также находит самую длинную увеличивающуюся подпоследовательность
- Stooge sort — рекурсивный алгоритм сортировки с временной сложностью $O(n^{\log_{1,5} 3}) \approx O(n^{2.71})$.
- Поразрядная сортировка — Сложность алгоритма: O(n·k); требуется O(k) дополнительной памяти. **Беркович**
- Цифровая сортировка — то же, что и Поразрядная сортировка.

Непрактичные алгоритмы сортировки

- Bogosort — O(n·n!) в среднем. Произвольно перемешать массив, проверить порядок.
- Сортировка перестановкой — O(n·n!) — худшее время. Генерируются всевозможные перестановки исходного массива и для каждой осуществляется проверка верного порядка.
- Глупая сортировка (Stupid sort) — O(n^3); рекурсивная версия требует дополнительно O(n^2) памяти

- Bead Sort — O(n) or O($\sqrt{n}$), требуется специализированное аппаратное обеспечение
- Блинная сортировка (Pancake sorting) — O(n), требуется специализированное аппаратное обеспечение

Алгоритмы, не основанные на сравнениях

- Блочная сортировка (Корзинная сортировка, Bucket sort)
- Лексикографическая или поразрядная сортировка (Radix sort)
- Сортировка подсчётом (Counting sort)

Остальные алгоритмы сортировки

- Топологическая сортировка
- Внешняя сортировка

См. также

- О-большое
- Временная сложность алгоритма
- Ёмкостная сложность алгоритма
- Внешняя сортировка
- Сортирующие сети (сравнение)
- Сравнивание
- Трансформация Шварца
- Параллельная сортировка
- Индексная сортировка

Предлагаемые для реализации методы сортировки:

1. **обменная сортировка** с разделением: алгоритм Ломуто («быстрая сортировка») [2 стр. 165]
2. **обменная сортировка** (метод распределяющего подсчета.) [1 стр. 445];
3. **обменная сортировка со слиянием (**сортировка Бэтчера)
4. **сортировка посредством выбора** (метод простого выбора)
5. **сортировка посредством выбора** (метод квадратичного выбора) [1 стр. 446]
6. **сортировка посредством выбора** (метод выбора с использованием бинарных деревьев; [1 стр. 446]
7. **сортировка посредством выбора** (метод “турнира с выбыванием”) [1 стр. 446]
8. **сортировка посредством выбора** (пирамидальная сортировка [1 стр. 446]
9. **сортировка Шелла** (сортировка с убывающим шагом) [1 стр. 465]
10. **сортировка вставками:** метод хеширования (сортировка с вычислением адреса) [1 стр. 465]
11. **сортировка вставками** (метод простых вставок)
12. **сортировка вставками** (метод двухпутевых вставок) [1 стр. 465]
13. **метод поразрядной сортировки** [1 стр. 465]) [2 стр. 173])
14. **сортировка с помощью кучи,** моделируя очереди с приоритетами на базе кучи. [2 стр. 147]
15. **сортировка вычерпыванием** [2 стр. 175]
16. **сортировка простым двухпутевым слиянием** [3 стр. 198].
17. **сортировка бинарного слияния** [3 стр. 246].