Задания к лабораторной работе №4 по дисциплине АиСД

Пониженная сложность, средняя сложность, повышенная сложность
В сложности учитывается не только сложность самого́ алгоритма, но и:
- рассматривается ли алгоритм на лекциях;
- нужно ли самостоятельно придумывать алгоритм.

Ниже для некоторых вариантов приведены ссылки на описания алгоритмов под соответствующими номерами.

Сортировки

Простые методы

1. Сортировка выбором; сортировка выбором с одновременным выбором максимума и минимума.
2. Сортировка простыми вставками; сортировка простыми вставками в список.
3. Двухпутевая сортировка бинарными вставками (при каждой вставке перемещаются не более половины элементов отсортированной последовательности).
4. Пузырьковая сортировка оптимизированная; сортировка чёт-нечет.
5. Гибрид сортировки пузырьком и сортировки выбором; шейкерная сортировка.
6. Бинго-сортировка.
7. Цикличная сортировка.

Усовершенствованные методы

8. Быстрая сортировка, рекурсивная реализация. Во время сортировки массив должен быть в состоянии:
   элементы <x, неотсортированные элементы, элементы >=x.
9. Быстрая сортировка, рекурсивная реализация. Во время сортировки массив должен быть в состоянии:
   элементы <x, элементы >=x, неотсортированные элементы.
10. Быстрая сортировка, рекурсивная реализация. Процедура трёхчастного разделения. Деление производится не на две группы, а на три: <x, =x, >x.
11. Быстрая сортировка, рекурсивная реализация – отсечение малых подмассивов.
    Быстрая сортировка выполняется не до конца. Когда сегменты становятся достаточно маленькими, они окончательно сортируются другим методом. Параметр, определяющий размер малого подмассива, задаётся пользователем.
12. Быстрая сортировка – итеративная реализация.
13. Пирамидальная сортировка.
14. Сортировка массивов слиянием – простое слияние, рекурсивная реализация.
15. Сортировка массивов слиянием – простое слияние, итеративная реализация.
16. Сортировка массивов слиянием – естественное слияние.
17. Нитевидная сортировка.
18. Сортировка Шелла.

19. Сортировка расчёской.
20. Поразрядная сортировка.
21. Соломонова сортировка.
22. Пасьянсная сортировка.
23. J-сортировка, 2 варианта:
- обычная J-сортировка;
- с многократным построением пары противоположных пирамид: сначала на всём массиве, потом на половине элементов в центре, потом на четверти элементов в центре и т.д. В конце – сортировка вставками на всём массиве.
24. Битонная сортировка (последовательная реализация).

Порядковые статистики и медиана.

25. k-й минимум – через пирамиду («малое» k ($k \le n$ / $\log_2 n$)).
26. k-й минимум – на основе разделения Хоара.
27. k-й минимум – линейный по сложности алгоритм на основе разделения по медиане медиан пятёрок.

Применение сортировок

28. Заданы два множества (размеров n и m). Являются ли они пересекающимися? Решить задачу, опираясь на сортировку (одного или обоих) множеств и бинарный поиск, если надо. Рассмотреть случай $m<<n$.
29. Заданы два множества (размеров n и m). Являются ли они пересекающимися? Решить задачу посредством *хеширования*. Создаётся хеш-таблица, содержащая элементы обоих множеств, и проверяется, что коллизии являются результатом хеширования идентичных элементов. На практике это может быть лучшим решением.
30. В наборе S имеется n вещественных чисел. Задано также вещественное число x. Содержатся ли в S два таких элемента, что их сумма равна x. Указание. Если набор S отсортирован, то решить задачу можно за время $O(n)$.
31. Заданы k отсортированных списков с общим количеством n элементов. Требуется осуществить их слияние в один отсортированный список. Указание. Простейший алгоритм требует времени $O(kn)$. Использовать пирамиду и получить время выполнения $O(n\log k)$.
32. «Болты и гайки». Имеется куча перемешанных n болтов и n гаек, отличающихся диаметрами. Нужно *быстро* найти все соответствующие пары болтов и гаек.
Известно, что каждая гайка подходит по диаметру ровно к одному болту, и наоборот. Нет ни двух болтов одинакового диаметра, ни двух гаек одинакового диаметра.
Попробовав навинтить гайку на болт, можно определить, что из них больше (или они соответствуют друг другу), но невозможно (а в алгоритме – нельзя) непосредственно сравнить два болта или две гайки.
Простой алгоритм – $O(n^2)$. Это *не быстро*. Предложить аналог рандомизированной быстрой сортировки – $O(n \log n)$.

Ссылки.

3. Д. Э. Кнут. Искусство программирования: Сортировка и поиск, Том 3. 5.2.1.
4. http://algolab.valemak.com/bubble-opt , http://algolab.valemak.com/odd-even
5. Шейкерная: http://algolab.valemak.com/cocktail
6. https://habr.com/ru/post/422085/
7. https://habr.com/ru/post/422085/
16. https://intuit.ru/studies/courses/648/504/lecture/11473?page=2
17. http://algolab.valemak.com/strand
18. https://ru.wikipedia.org/wiki/Сортировка_Шелла
19. http://algolab.valemak.com/comb
21. http://algolab.valemak.com/solomon
22. https://habr.com/en/company/edison/blog/431094/
23. https://habr.com/ru/post/221095/
24. https://ru.wikipedia.org/wiki/Битонная_сортировка#Описание_алгоритма (перед сортировкой дополнить массив фиктивными элементами до степени двойки)