

СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ

сверстал какой-то Никита Батькович

2021-2022

Оглавление

Глава 1

Основные принципы

Глава 2

Структуры данных

2.1 Связные списки

2.2 Стеки

2.3 Очереди

2.4 Деревья

2.5 Графы

Глава 3

Алгоритмы

3.1 Сортировки

3.1.1 Вставкой

3.1.2 Пузырьковая

3.1.3 Шейкерная

Эта сортировка является улучшением пузырьковой:

1. Запомнить, были или не были перестановки в процессе очередного прохода, если в последнем проходе перестановок не было, то алгоритм можно закончить.

2. Запомнить, не только сам факт, что обмен имел место, но и индекс последнего обмена.

3. Ясно, что все пары после этого индекса уже отсортированы.

4. Проводить сортировку последовательно в двух направлениях по массиву.

В лучшем случае, на уже отсортированном массиве сложность метода $O(n)$, однако, в

худшем (в обратном порядке отсортированном) - $O(n^2)$

3.1.4 Шелла

Сначала отдельно группируются и сортируются элементы отстоящие друг от друга на расстоянии $n/2$, затем $n/4$ и т.д. пока не будет выполнена обычная одинарная сортировка. На каждом проходе сортировка программируется как сортировка прямым включением или вставками.

На каждом этапе либо сортируется относительно мало элементов либо элементы уже достаточно хорошо отсортированные и требуется немного перестановок.

Расстояние в группах можно уменьшать по разному, единственное требование, чтобы всё завершалось одинарной сортировкой, где в самом худшем случае и будет сделана вся работа. Окончательно неизвестно какие именно расстояния дают наилучший результат, но точно не степени двойки и не множители друг другу.

Неплохие результаты дают последовательности 1, 4, 13, 40, 121... Один из лучших считается 1,4, 10, 23, 57, 132, 301, 701... В последнем случае сложность $O(n^*1,3)$

3.1.5 Пирамидальная сортировка

Одномерный массив представляется в виде бинарного дерева (пирамиды). В классике нумерация массива берётся с 1 и левый потомок располагается под номером $2*i$, а правый $2*i+1$.

ТУТ ПРИМЕР С ДЕРЕВОМ РИСУНОК

Данное дерево перестраивается таким образом, чтобы потомок никакого узла не был больше своего родителя. В результате на первом месте окажется наибольший элемент, поменяем его местами с последним и дерево будет почти хорошим за исключением первого элемента.

Остаётся "протолкнуть" единицу по дереву на своё место и т.д.

При проталкивании по дереву каждый элемент проходит не более $\log_2 n$ в основании 2, таких элементов n , которые мы проталкиваем. Значит, сложность сортировки $O(n \log_2 n)$

3.2 Классификация сортировок

3.3 Поиска

3.4 На графах

3.5 Обработки строк