

Лабораторная работа №10

Анализ алгоритмов улучшенных методов сортировки массивов

Цель работы: Закрепление знаний алгоритмов сортировки. Приобретение навыка анализа и оценки алгоритмов улучшенных методов сортировки.

Подготовка к работе : Изучить алгоритмы улучшенных методов сортировки и провести анализ и оценки алгоритмов улучшенных методов сортировок.

Контрольные вопросы

1. В чем заключается особенность алгоритма Шелла? Опишите суть алгоритма.
2. В чем заключается особенность алгоритма быстрой сортировки? Опишите суть алгоритма.
3. В чем заключается алгоритм пирамидальной сортировки? Опишите суть метода
4. В чем заключается сортировка разделением? Опишите суть метода.
5. В чем заключается алгоритм сортировки слиянием? Опишите суть метода

Задание

Выполнить исследование улучшенных методов сортировки (алгоритм Шелла, алгоритм быстрой сортировки, алгоритм пирамидальной сортировки, radix) для числового массива и связанного списка (по тем же данным) по времени выполнения и указанным параметрам. (мощность массива $n=10, 50, 100, 1000, 10000, 100000$).

1. Провести анализ проведенных расчетов для лучшего, среднего и худшего случаев для всех улучшенных алгоритмов сортировки массива. Для всех видов сортировок , для структур данных (массив, статический список односвязный и двусвязный список) использовать одни и те же подготовленные заранее данные, записанные в файлы. Численный эксперимент проводить для лучшего случая (данные отсортированы в заданном порядке), для среднего случая (случайным образом сформированы данные), для худшего случая (данные отсортированы в обратном порядке). Для мощностей-
 - ✓ N - 10, 50 - диапазон случайных данных- -100 до 100;
 - ✓ N - 100 - -1000 до 1000;
 - ✓ N -1000 - 10000 до 10000.

И так далее

2. Провести анализ проведенных расчетов для лучшего, среднего и худшего случаев для всех улучшенных алгоритмов сортировки односвязного списка. (для тех же данных и параметров). Параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных. Обоснованный вывод по результатам анализа.
3. Провести анализ проведенных расчетов для лучшего, среднего и худшего случаев для всех улучшенных алгоритмов сортировки двусвязного списка (для тех же данных и параметров). Параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных. Обоснованный вывод по результатам анализа.

Результаты анализа представить в виде таблицы Excel и графиков-

- ✓ Представить выводы

Пример

1. Сортировка шелла

Сложность для лучшего случая – $O(n \cdot \log n)$

Сложность для среднего случая – $O(n \cdot \log n)$

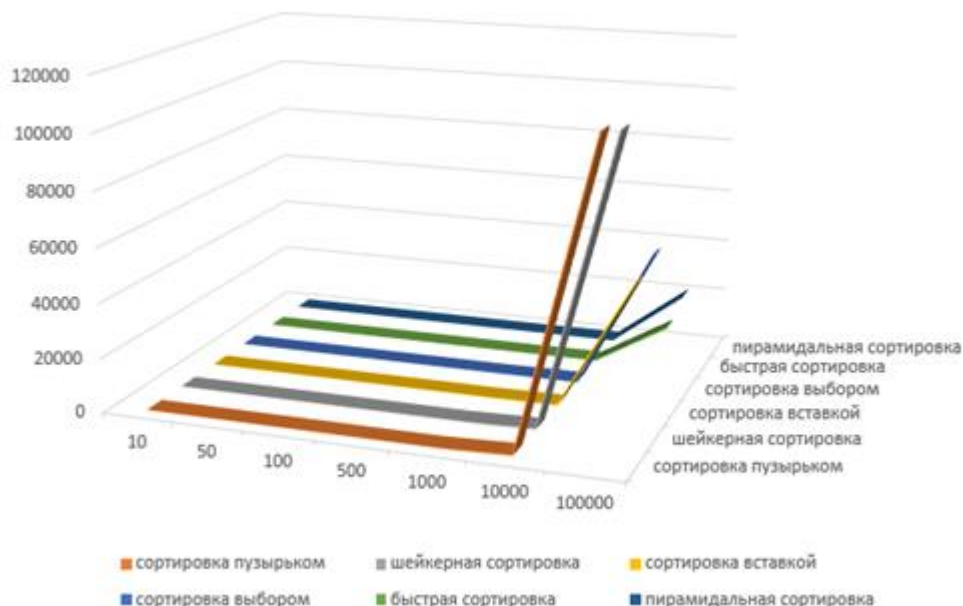
Сложность для худшего случая – $O(n^3)$

Время, кол-во сравнений и перестановок для сортировки шелла (массив)								
кол-во элементов:	10	50	100	500	1000	10000	100000	
Время	0	0	0	0	0	0	0	лучший случай
сравнения	22	203	503	3506	8006	120005	1500006	
перестановки	0	0	0	0	0	0	0	
Время	0	0	0	0	0	0	46	средний случай
сравнения	31	359	861	6409	14913	259248	4167991	
перестановки	14	185	420	3175	7419	144283	2718400	
Время	0	0	0	0	0	0	0	худший случай
сравнения	27	263	668	5116	11716	172578	2244585	
перестановки	13	105	260	2100	4700	62560	844560	

Время, кол-во сравнений и перестановок для сортировки шелла (статический список)								
кол-во элементов:	10	50	100	500	1000	10000	100000	
Время	0	0	0	0	0	0	15	лучший случай
сравнения	22	203	503	3506	8006	120005	1500006	
перестановки	0	0	0	0	0	0	0	
Время	0	0	0	0	0	16	94	средний случай
сравнения	31	359	861	6409	14913	259248	4167991	
перестановки	14	185	420	3175	7419	144283	2718400	
Время	0	0	0	0	0	15	47	худший случай
сравнения	27	263	668	5116	11716	172578	2244585	
перестановки	13	105	260	2100	4700	62560	844560	

Из анализа четырех предыдущих алгоритмов мы можем сделать вывод: чем меньше элементу нужно пройти до его позиции в отсортированном массиве – тем меньше выполняется перестановка и тем меньше нужно сделать сравнений. Первые четыре перемещают каждый элемент только на одну позицию при каждом элементарном шаге. В итоге таких шагов все равно приходится делать приблизительно n^2 . Чтобы сделать серьезное улучшение производительности – необходимо найти способ передвигать элементы более чем на одну позицию за элементарный шаг. В связи с описанным становится понятно, почему сортировка шелла не эффективна для связанного списка: данная сортировка требует возможности получать доступ к любому элементу в последовательности за константное время. Данный метод является улучшением

Связный список (средний случай)



Время, кол-во сравнений и перестановок для всех сортировок (связный список, лучший случай)								
кол-во элементов:	10	50	100	500	1000	10000	100000	
Время	0	0	0	0	0	0	0	сортиров
сравнения	9	49	99	499	999	9999	99999	ка
перестановки	0	0	0	0	0	0	0	пузырьк
Время	0	0	0	0	0	0	0	шейкерн
сравнения	9	49	99	499	999	9999	99999	ал
перестановки	0	0	0	0	0	0	0	сортиров
Время	0	0	0	0	0	0	0	сортиров
сравнения	9	49	99	499	999	9999	99999	ка
перестановки	0	0	0	0	0	0	0	вставкой
Время	0	0	0	0	0	266	29688	сортиров
сравнения	36	1176	4851	124251	498501	49985001	4999850001	ка
перестановки	9	49	99	499	999	9999	99999	выбором
Время	0	0	0	0	0	109	13312	быстрая
сравнения	31	255	606	4008	9009	125439	1600016	сортиров
перестановки	6	31	63	255	511	5904	65535	ка
Время	0	0	0	0	16	156	19563	пирамид
сравнения	119	901	2081	13789	30470	405077	5040128	альная
перестановки	31	257	616	4176	9316	126696	1597434	сортиров