Московский авиационный институт
(государственный технический университет)

Факультет «Прикладная математика и физика»
Кафедра «Вычислительная математика и информатика»

Курсовой проект по

Языкам и методам программирования по теме:

«Сортировка и поиск»

Выполнил: Щербаков А.А.
Студент гр. М8О-106Б
Преподаватель: Дубинин А.В.
Оценка:
Дата:

2017 год

Введение

Человеку неудобно работать с информацией, расположенной в хаотичном порядке. Возможность представить эту информацию в сортированном виде позволяет ускорить работу с ними. В случае обработки информации компьютером такая возможность становится необходимостью. Такое представление предоставляет некоторые преимущества, например, более быстрый поиск необходимых кусочков информации. Также есть и большой минус – сортировка различных данных требует большого количества операций. Таким образом, необходимо учитывать, действительно ли сортировка данных принесет заметный результат при последующей работе с ними или нет.

Методы сортировки

Существует несколько основных наиболее часто используемых методов сортировки:

1. Линейный выбор с обменом – O(n^2)
2. Линейный выбор с подсчетом – O(n^2)
3. Метод пузырька – O(n^2)
4. Шейкер сортировка – O(n^2)
5. Метод простой вставки – O(n^2)
6. Метод двоичной вставки – O(n^2)
7. Метод Шелла – O(n*log(n))
8. Турнирная сортировка – O(n*log(n))
9. Пирамидальная сортировка с просеиванием – O(n*log(n))
10. Простое двухпоточное слияние – O(n*log(n))
11. Быстрая сортировка Хоара – O(n*log(n)~n^2)
12. Четно-нечетная сортировка – O(n^2)
13. Прямое слияние – O(n*log(n))
14. Естественное слияние – O(n*log(n))
15. Гладкая сортировка – O(n*log(n))

Мой метод сортировки — сортировка Шелла.

Сортировка Шелла

При сортировке Шелла сначала сравниваются и сортируются между собой значения, стоящие один от другого на некотором расстоянии. После этого процедура повторяется для некоторых меньших значений, а завершается сортировка Шелла упорядочиванием элементов при (то есть обычной сортировкой вставками). Эффективность сортировки Шелла в определённых случаях обеспечивается тем, что элементы «быстрее» встают на свои места (в простых методах сортировки, например, пузырьковой, каждая перестановка двух элементов уменьшает количество инверсий в списке максимум на 1, а при сортировке Шелла это число может быть больше).

Невзирая на то, что сортировка Шелла во многих случаях медленнее, чем быстрая сортировка, она имеет ряд преимуществ:

- отсутствие потребности в памяти под стек;
- отсутствие деградации при неудачных наборах данных — быстрая сортировка легко деградирует до $O(n^2)$, что хуже, чем худшее гарантированное время для сортировки Шелла.

В методе Шелла сравниваются элементы, расположенные на расстоянии d (где d — шаг между элементами, которые сравниваются). Если d = [n/2], то после каждого просмотра шаг d уменьшается вдвое. На последнем просмотре он сокращается до d=1.

Например, пусть дан список, в котором число элементов четно:{40, 11, 83, 57, 32, 21, 75, 64}. Список длины n разбивается на n/2 частей, т. е. d = [n/2] = 4, где [ ] — целая часть числа.

При первом просмотре сравниваются элементы, отстоящие друг от друга на d=4 (шаг d = 4), т. е. K1 и k5, k2 и k6 и т.д. Если ki>ki+d, то происходит обмен между позициями i и (i+d).

Исходный массив	40	11	83	57	32	21	75	64
Шаг $d = 4$	32	11	75	57	40	21	83	64
Полученный массив	32	11	75	57	40	21	83	64

Перед вторым просмотром выбирается шаг d=[d/2]= 2 (шаг d= 2).

Исходный массив	32	11	75	57	40	21	83	64
Шаг $d = 2$	32	11	40	21	75	57	83	64
Полученный массив	32	11	40	21	75	57	83	64

Затем выбираем шагd=[d/2]= 1 (рис 10 – Метод Шелла (шаг d= 1)), т.е. имеем аналогию с методом стандартного обмена.

Исходный список	32	11	40	21	75	57	83	64
Шаг $d = 1$	11	32	32	40	21	40	40	75	57	75	75	83	64	83
Полученный список	11	32	21	40	57	75	64	(83)

Сложность метода Шелла ***O(n*log(n)).***

Заключение

Действительно, путем сортировки данных по некоторым ключам ощутимо меняется последующая скорость работы с ними. Например, если необходимо много раз искать подходящий элемент среди данных, то поддерживание этих данных в сортированном виде в среднем ускоряет этот процесс. Оптимальным алгоритмом для поиска заданного элемента в массиве является бинарный поиск, который в худшем случае работает за O(log(n)). По сравнению с линейным поиском в несортированном массиве скорость работы заметно отличается. В случае небольшого количества бинарных поисков(< n) алгоритм сортировки за O(n^2) теряет смысл. Таким образом, сортировка является важнейшим методом обработки информации. Правильный выбор метода сортировки (устойчивый, неустойчивый, зависимый от характера входных данных) позволяет добиться более быстрой работы программ.

Списки источников

1. ru.wikipedia.org/wiki/Алгоритм_сортировки Стандартные методы сортировки;

2. https://habrahabr.ru/post/133996/ - Сортировки;

3.http://algol.adept-proekt.ru/algoritmi-sortirovri - Алгоритмы сортировки.