А. П. Купрейчик, Д. С. Панкевич

ПОИСК УНИВЕРСАЛЬНОЙ ЛИНЕЙНОЙ СОРТИРОВКИ

Описываются линейные алгоритмы сортировки, целесообразность их использования и преимущества перед другими методами сортировки

Сортировки и их важность

Сегодня сортировки используются повсеместно. Огромные базы данных постоянно должны быть отсортированы для различных целей – как для пользователей, так и для создателей. Скорость работы сортировки очень важна для больших объемов данных – никто не хочет ждать долгое время, пока его база данных сортируется.

I. Сортировки, основанные на сравнении и их лимит скорости в O(n*log(n)).

Первые сортировки были основаны на сравнении элементов. Они легко оперируют всеми типами данных, у которых определено сравнение. Однако доказано, что ни одна из сортировок со сравнениями не может сработать быстрее, чем за O(n*log(n)), где n – количество элементов. Такие сортировки при средней скорости процессора в 10^8 операций в секунду отсортирует около 40.000.000 элементов за секунду. Это большое число, но иногда его недостаточно.

II. Сортировки, основанные не на сравнении.

Чуть позже появились сортировки, основанные не на сравнении. Для них лимита в O(n*log(n)) нет – их скорость может достигать O(n), что немного быстрее. Первая из таких сортировок – сортировка блоками, или корзинами. Ее суть в том, что элементы распределяются по конечному числу корзин (карманов, блоков) и сортируются внутри корзин, после чего сливаются обратно. При удачном выборе корзин сортировка обладает линейным временем.

Многие известные сортировки являются модификацией этой сортировки. Быстрая сортировка – это сортировка корзиной, но с двумя коризнами. При этом элементы в корзину попадают, основываясь на сравнении с элементом-корнем, выбранным ранее.

Еще одна сортировка – поразрядная. Ее суть – мы по очереди сортируем массив по каждому разряду чисел, после чего массив оказывается отсортированным. Ее минусы – работа только с целыми числами и наличие делений, что сильно замедляет работу программы.

Сортировка подсчетом основывается на подсчете количества элементов и размещении их сразу на правильном месте. Это модификация сортировки корзиной, с количеством корзин, равным максимальному элементу.

III. Этапы модификации поразрядной сортировки

Мы решили улучшить поразрядную сортировку, увеличив ее скорость работы и возможности. Одна из ее проблем – это деление для получения разряда числа. Деление – медленная операция, и мы решили заменить ее сдвигом и сортировать по байтам, что в 4 раз ускорило сортировку для большого количества элементов. Также, мы модифицировали нашу байтовую сортировку сортировкой подсчетом, что еще немного ускорило ее работу.

После мы занялись решением проблемы с типами данных, так как нецелые типы данных по умолчанию не поддерживают операции битовых сдвигов. Эта проблема была решена использованием конструкций union и шаблонностью типов данных с плавающей запятой (например, у типа float биты всегда стоят в одинаковом порядке: один бит на знак, 8 на порядок и 23 на мантиссу). Отсортировав по очереди каждый элемент числа, мы получим отсортированный массив.

IV. Выводы

Мы получили универсальную линейную сортировку, которая почти во всех случаях лучше самых быстрых сортировок, основанных на сравнении (кроме типов данных double и long double). Мы считаем, что использование линейных сортировок должно быть внедрено повсеместно (например, в компиляторе gcc по умолчанию используется сортировка слияниемвставкой, которая в разы медленнее нашей). Поэтому наш проект нацелен на ускорение работы всех систем СУБД и любых программ, сортирующих данные.

1. Pierre Terdiman. Radix Sort Revisited / Pierre Terdiman /https://codercorner.com/RadixSortRevisited.htm – 04.01.2000

Купрейчик Артём Павлович, Панкевич Даниил Сергеевич, учащиеся факультета информационных технологий и управления, БГУИР,antanta1907@mail.ru

Научный руководитель: Кривоносова Татьяна Михайловна, доцент кафедры вычислительных методов и программирования БГУИР, krivonosova@bsuir.by.