УДК 372

Михасик Е. И.

ЭФФЕКТИВНОСТЬ АЛГОРИТМА СОРТИРОВКИ

БНТУ, г. Минск
Научный руководитель: канд. техн. наук, доцент Дробыш А. А.

Любая компьютерная программа подразумевает набор алгоритмов, которые позволяют решить те или иные поставленные задачи. Алгоритм – это конечная совокупность точно заданных правил решения произвольного класса задач или набор инструкций, описывающих порядок действий исполнителя для решения конкретной задачи. Одной из основных и наиболее часто используемых задач, как промежуточных, так и итоговых, является задача сортировки элементов массива или множества. Существует огромное количество методов сортировки, некоторые известны каждому начинающему программисту (например, сортировка пузырьком или сортировка выбором), а о некоторых не догадываются даже умудренные практической деятельности асы программирования (например, гномья сортировка, сортировка Шелла). Все они отличаются собственно самим набором операций, которые выполняются для получения результата, а также необходимыми для этого ресурсами. Соответственно, для оценки эффективности алгоритма сортировки необходимо оценить следующие параметры: время выполнения; затраты оперативной памяти; громоздкость алгоритма.

Время выполнения – оценка быстродействия алгоритма, Показатель, определяющий сколько времени потребуется для выполнения всего алгоритма. Если в алгоритм подается множество из n элементов, то алгоритм считается наиболее оптимальным, если на его выполнение уходит O(n log n) времени, плохой алгоритм – $O(n^2)$, идеальный показатель составляет – O(n), однако добиться его чрезвычайно сложно и при сложности проведения операций невозможно. O – это некая константа, которая подразумевает, что при увеличении количества элементов, над которыми производятся некоторые операции, время осуществления операции будет изменяться не более, чем на некоторую константу, асимптоту.

Память – оценка дополнительной памяти, которую необходимо выделить под временное хранение данных.

Громоздкость алгоритма – количество операций и преобразований, необходимых для выполнения сортировки.

Рассмотрим основные виды сортировок и показатели их эффективности.

Сортировка пузырьком. Данный вид сортировки подразумевает следующие действия – проверяется массив элементов слева направо и, если текущий элемент больше следующего, то они меняются местами. После первой итерации самый большой элемент будет находиться на своем месте. Получается, что для выполнения всей сортировки, нам потребуется не более n итераций. В лучшем случае асимптотика составляет O(n), в худшем она будет приближаться к $O(n^2)$. Данный алгоритм не требует лишнего дополнительного выделения памяти, также сам алгоритм, например, на языке C++ сортировка одномерного массива занимает порядка 10 строк с вложенными условными операторами. На данный момент он является наиболее распространенным алгоритмом сортировки для небольших массивов элементов.

Сортировка вставками. Данный вид сортировки подразумевает создание еще одного массива, в котором будет находиться отсортированный исходный массив. После каждой итерации мы будем добавлять в новый массив элемент, получается, что для завершения сортировки необходимо n итераций. В лучшем случае асимптотика составляет O(n), в худшем случае она будет близка к $O(n^2)$. Однако, для данного метода необходима дополнительная память для хранения еще одного массива. Даже несмотря на всю простоту процесса и не громоздкости алгоритма, при сортировке большого по размеру массива (например, массив строковых переменных) необходимо выделять память под такой же по размеру массив и это становится неэффективно.

Сортировка выбором. В данном случае находится минимальный элемент в массиве после данного элемента и меняется с ним, если надо. Соответственно, после каждой итерации мы получаем один элемент на своем месте. Но так как массив проходится каждый раз от начала до конца, то асимптотика составляет $O(n^2)$. Он подразумевает работу только с небольшими массивами, так как постоянный просмотр и проверка массива нагружает процессор, и работа с большими массивами данных становится неудобной и долгой.

Сортировка расческой. Она подразумевает собой две сортировки в одной. Сначала выбирается некий промежуток в массиве, удобнее всего весь массив, и сравниваем их между собой. Если необходимо, то меняем их местами. Далее промежуток уменьшаем, в общем случае исходное расстояние делим на 1,3 (при работе большими массивами 1,247), и проверяем на этом расстоянии элементы массива и движемся до конца массива. Так продолжается до того момента, пока шаг не будет равен 1, после этого проводим сортировку пузырьком. В лучшем случае время вычисляется по формуле O(n log n). В худшем случае $O(n^2)$. Данный метод является сложным по написанию в программе, однако, он помогает не просматривать каждый раз весь массив данных, а только в самом конце при сортировке пузырьком. Поэтому данный метод является наиболее эффективным для работы с большими массивами данных.

Шейкерная сортировка. Модифицированная сортировка пузырьком, которая ограничивает неотсортированные элементы массива. Для этого вводятся специальные ограничители начала и конца, которые сдвигаются после каждой итерации. Это позволяет улучшить скорость работы алгоритма и приблизиться к идеальному показателю O(n). Но при большом массиве все равно время приближается к $O(n^2)$. Поэтому в больших массивах данная сортировка неуместна, несмотря на то, что описание ее в программе занимает небольшой объем.

Кроме этих видов сортировок существует огромное количество других, которые позволяют сортировать как больших, так и маленькие массивы численных и строковых элементов, однако, все они являются узкоспециализированные и ни одна из них не может показать идеальную асимптотику O(n) при любом значении n. Значит, каждый программист волен сам выбирать для себя метод сортировки, который наилучшим образом подходит для решения конкретной задачи.