Тема 5. Сортировка

5.5. Распределяющая сортировка

Распределяющая сортировка основана на том, что имена распределяются по группам и содержимое групп затем объединяется таким образом, чтобы частично отсортировать таблицу; процесс повторяется до тех пор, пока таблица не будет отсортирована полностью. Такая сортировка отличается от ранее рассмотренных алгоритмов тем, что она основана не на сравнении имен, а на их представлении. Предполагается, что каждое из имен $x_1, x_2, \ldots, x_n$ имеет вид $x_i = (x_{i,p}, x_{i,p-1}, \ldots, x_{i,1})$ и их необходимо отсортировать в возрастающем лексикографическом порядке, т. е.

$$x_i = (x_{i,p}, x_{i,p-1}, \ldots, x_{i,1}) < x_j = (x_{j,p}, x_{j,p-1}, \ldots, x_{j,1})$$

тогда и только тогда, когда для некоторого $t \le p$ имеем $x_{i,l} = x_{j,l}$ для $l > t$ и $x_{i,t} < x_{j,t}$. Для простоты будем считать, что $0 \le x_{i,l} < r$, и поэтому имена можно рассматривать как целые, представленные по основанию r, т. е. каждое имя состоит из p r-ичных цифр. Для того чтобы не было имен разной длины, более короткие имена дополняются нулями. Поэтому такая сортировка часто называется *цифровой распределяющей сортировкой*.

Цифровая распределяющая сортировка основана на том, что если имена уже отсортированы по младшим разрядам l , $l - 1$, …, 1, то их можно полностью отсортировать, сортируя только по старшим разрядам p, $p - 1$, …, $l + 1$ при условии, что сортировка осуществляется таким образом, чтобы не нарушить относительный порядок имен с одинаковыми цифрами в старших разрядах. Таким образом, поразрядную сортировку можно выполнить следующим образом: сначала произвести распределяющую сортировку по младшему разряду имен, переместив имена во вспомогательную область, затем произвести еще одну распределяющую сортировку по следующему разряду, переместив предварительно имена обратно в исходную область памяти, и так до тех пор, пока таблица не будет отсортирована.

Для реализации распределяющей сортировки лучше использовать связные структуры данных. В этом случае не придется перемещать имена, а достаточно скорректировать связи (указатели). Алгоритм 5.9 представляет общую процедуру цифровой распределяющей сортировки. Исходная таблица x_1, x_2, …, x_n организуется в очередь Q. Организуется r пустых очередей Q_0, Q_1, …, Q_{r-1} по одной для каждого целого числа от 0 до $r - 1$.

Сформировать из $x_1, x_2, \ldots, x_n$ входную очередь Q

for $j \leftarrow 1$ **to** p **do**
- Сделать очереди $Q_0, Q_1, \ldots, Q_{r-1}$ пустыми
- **while** Q не пуста **do**
  - $X \Leftarrow Q$
  - Пусть $X = (x_p, x_{p-1}, \ldots, x_1)$
  - $Q_{x_j} \Leftarrow X$
- Сцепить очереди $Q_0, Q_1, \ldots, Q_{r-1}$ вместе для формирования новой очереди Q

Алгоритм 5.9. Цифровая распределяющая сортировка

Просматривается последовательность $x_1, x_2, \ldots, x_n$ слева направо и элемент x_i помещается в ту очередь, которая соответствует цифре младшего разряда. Как только все имена распределены по соответствующим очередям, очереди $Q_0, Q_1, \ldots, Q_{r-1}$ объединяются вместе в порядке возрастания в очередь Q. Объединение очередей подразумевает их сцепление, т. е. содержимое $(l + 1)$-й очереди приписывается к концу l-й очереди (конкатенация очередей). Затем процесс повторяется для следующего разряда слева и так до тех пор, пока не будут отсортированы все имена по всем разрядам. В результате Q будет содержать имена в порядке возрастания, начиная с головы очереди Q. Процедура такой сортировки для трехзначных десятичных чисел показана на рис. 5.12.

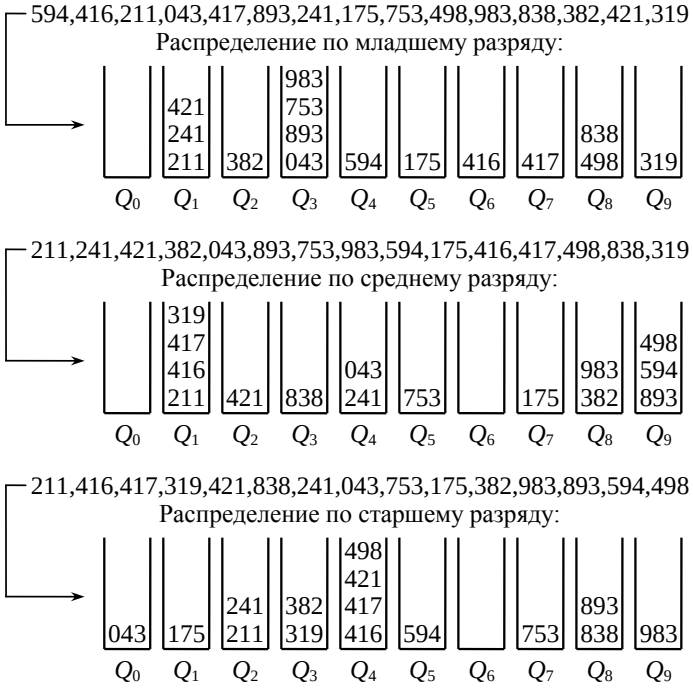

Рис. 5.12. Процесс цифровой распределяющей сортировки

Время работы алгоритма определяется общим числом операций с очередями. Всегда производится p проходов по таблице. На каждом проходе производится исключение из очереди Q каждого имени и включение его в одну из очередей Q_i. Таким образом, всего в очередях производится $2np$ операций включения/исключения. На каждом проходе выполняется также $r - 1$ сцеплений для получения очереди Q из очередей Q_0, Q_1, …, Q_{r-1}, т. е. всего имеется $(r - 1)p$ операций сцепления. Поскольку операции включения/исключения и операции сцепления могут быть выполнены за постоянное время (не зависящее от размера очереди), требуется всего $2np + (r - 1)p$ операций с очередями. Таким образом, алгоритм требует времени $O(np + rp)$. Для случаев, когда n существенно больше r, время работы алгоритма определяется в большей степени числом имен n, чем числом разрядов, и составляет $O(n)$, т. е. цифровая распределяющая сортировка может оказаться одним из наиболее эффективных методов внутренней сортировки.

Для повышения эффективности цифровой распределяющей сортировки можно использовать компромиссный подход, который использует распределяющую сортировку только по нескольким старшим разрядам, а затем применяется простая сортировка вставками. Эффект достигается за счет того, что после распределяющей сортировки по старшим цифрам таблица становится почти упорядоченной (т. е. существенно уменьшается число инверсий), и простая сортировка вставками становится эффективной.