

## SECTION XIX. COMPUTER AND SOFTWARE ENGINEERING

DOI 10.36074/logos-26.02.2021.v2.03

### ПРОЦЕСС РАНЖИРОВАНИЯ ОБЪЕКТОВ И ЕГО РЕАЛИЗАЦИЯ

**Мартынюк Татьяна Борисовна**

доктор технических наук, профессор,  
профессор кафедры вычислительной техники  
*Винницкий национальный технических университет*

**Круковский Богдан Игоревич**

аспирант кафедры вычислительной техники  
*Винницкий национальный технических университет*

УКРАИНА

**Введение.** Ранжирование числовых данных в качестве объектов цифровой обработки является специфической операцией. Ее реализация связана с сортировкой числовой последовательности и является ее результатом, который позволяет сделать более представимым распределение чисел в отсортированном массиве, например, в порядке возрастания их значений [1].

**Цель работы.** Целью является анализ особенностей реализации ранжирования числовых данных.

**Материалы и методы.** Наиболее перспективными областями применения результатов ранжирования числовых данных являются медианная фильтрация [2] и медицинская диагностика [3-5].

Процедура медианной фильтрации часто используется при предварительной обработке сигналов, а специфической особенностью медианных фильтров является избирательность по отношению к элементам массива, которые представляют собой немонотонную составляющую последовательности чисел в пределах «окна» фильтра и резко выделяются на фоне соседних отсчетов [2]. При этом два варианта реализации процесса медианной фильтрации изображений можно представить следующим образом (рис.1).

В результате, среднее значение отсчетов скользящего «окна» фильтра можно получить после их сортировки, а не в процессе вычисления среднего арифметического значения всех элементов (пикселей) этого «окна» (рис. 1). Такой подход значительно ускоряет процесс медианной фильтрации полутоновых изображений при определении значения центрального элемента «окна» апертуры.

При медицинском диагностировании заболеваний в некоторых случаях важно определить не только наиболее вероятный диагноз, который имеет наивысший ранг соответственно, но и ближайшие к нему диагнозы по значению их рангов [4, 5]. Все это позволяет в дальнейшем уточнить диагноз, расширив список анализируемых симптомов и выделив среди них значащие.

Таким образом, ранжирование объектов (не только числовых данных) позволяет выполнить их классификацию (диагностирование) в более широком

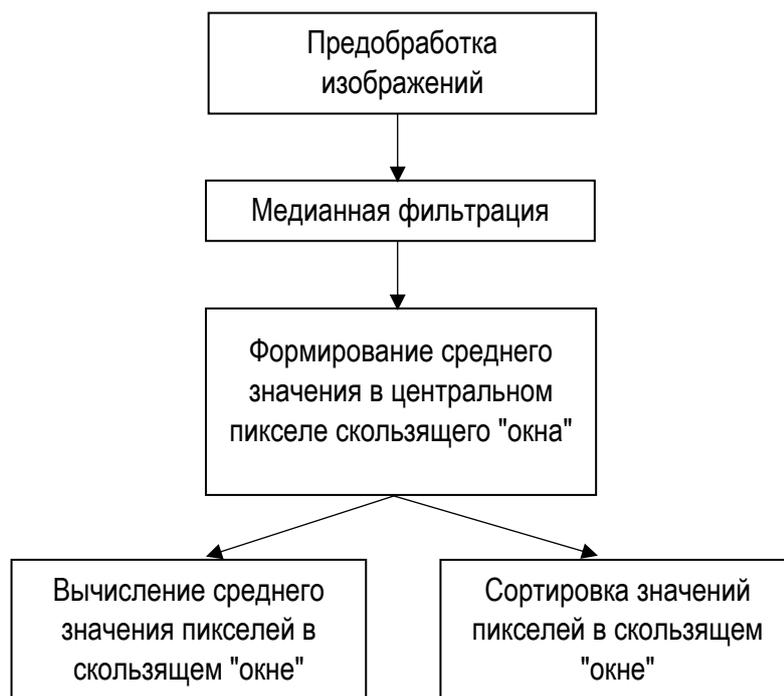


Рис. 1. Особенности реализации медианной фильтрации

виде с возможностью дальнейшего уточнения результатов, а также с возможностью наглядного представления картины распределения объекта по классам даже для случая, когда объект классификации представлен как совокупность детерминированных признаков [3].

Сам процесс ранжирования объектов представляет собой процедуру упорядочения, который формирует объекты в порядке их предпочтения, руководствуясь одним или несколькими выбранными показателями сравнения [1, 6]. Процедура ранжирования применяется в случаях, когда невозможна или нецелесообразна непосредственная оценка. При этом ранжирование объектов содержит только информацию о том, какой из объектов лучше, и не содержит информацию о том, насколько или во сколько раз один объект преобладает среди других [6].

Кроме того, использование рангов можно рассматривать как внутренний адрес, который позволяет считывать данные из ассоциативной памяти по их содержанию [7]. В любом случае ранжирование рассматривается как результат процесса сортировки.

В данной работе представлен способ сортировки одномерного (векторного) массива чисел, использующий векторный принцип обработки, т.е. как вариант обработки параллельно по словам, но последовательно по разрядам слов [7]. При этом использование операций декремента/инкремента позволяет, во-первых, отказаться от группы  $n/2$  компараторов, где  $n$  – размерность сортируемого массива чисел, а во-вторых, использовать группу  $n$  счетчиков в качестве памяти рангов [8].

На (рис. 2) представлена структурная схема вычислителя, который содержит сортировщик, регистровую память данных и память рангов [8]. Регистровая память используется для хранения массива чисел, который для сортировки переписывается в сортировщик, а память рангов используется для

фиксации сформированных рангов чисел массива. Основная функциональная нагрузка по сортировке выполняется сортировщиком, который содержит массив  $n$  счетчиков и анализатор реакций. Память рангов также содержит массив  $n$  счетчиков.

Параллельное сравнение чисел, зафиксированных в соответствующих счетчиках сортировщика, выполняется в процессе одновременного уменьшения их содержимого на единицу при каждом синхроимпульсе (операция декремента) до их последовательного обнуления. Поскольку первоначально во все счетчики памяти рангов были записаны значения единицы, то фактически всем числам массива был присвоен минимальный ранг 1.

Момент обнуления содержимого каждого счетчика сортировщика фиксируется анализатором реакций, который запрещает в этом цикле увеличение на единицу (операция инкремента) содержимого соответствующего счетчика памяти ранга. В остальных счетчиках памяти рангов их содержимое (ранги) увеличиваются на единицу.

После последнего обнуления содержимого всех счетчиков сортировщика процесс формирования рангов завершается. Сформированные ранги можно использовать для считывания из регистровой памяти соответствующих данных, а именно, максимальное / минимальное число или число со средним значением [8].

Особенностью предложенного способа сортировки с ранжированием (рис. 2) является то, что в процессе сортировки не используется попарное сравнения элементов массива чисел с последующей их перестановкой, а только операция декремента параллельно по всем числам. А это, в свою очередь, дает возможность одновременно сформировать ранг каждого из элементов массива чисел с использованием операции инкремента. Это позволяет определить в дальнейшем любой элемент входного массива по его рангу (местоположению) в отсортированном массиве.

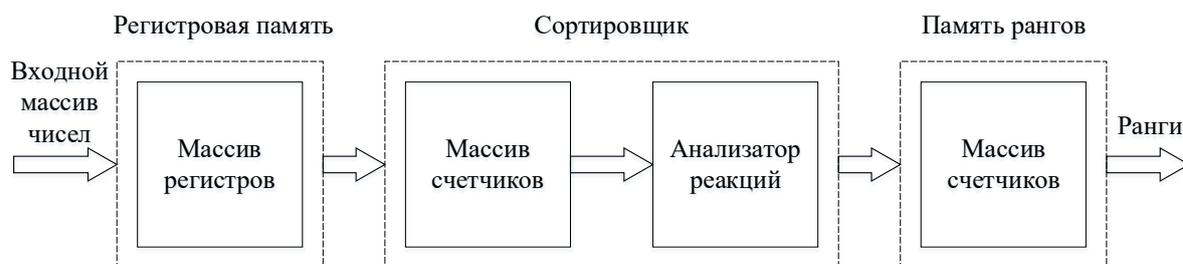


Рис. 2 Структурная схема вычислителя для сортировки с ранжированием

**Результаты и обсуждения.** Результат сортировки с ранжированием массива из четырех чисел {2, 5, 3, 4} представлен в (табл. 1). В нулевом цикле в массиве рангов фиксируется единица. В течение первого цикла элементы массива чисел уменьшаются на единицу (операция декремента), а элементы массива рангов, в свою очередь, остаются без изменений. На втором цикле один из элементов массива чисел обнуляется, а именно первый элемент. В результате его ранг остается неизменным (со значением 1), а ранги всех остальных чисел увеличиваются на единицу и принимают значение 2 (операция инкремента). Процесс сортировки происходит до тех пор, пока в массиве чисел все элементы не станут нулями, а в результате будет сформирован массив рангов.

Таблица 1

## Процесс сортировки с ранжированием

0-й цикл		1-й цикл		2-й цикл	
Массив чисел	Массив рангов	Массив чисел	Массив рангов	Массив чисел	Массив рангов
2	1	1	1	0	1
5	1	4	1	3	2
3	1	2	1	1	2
4	1	3	1	2	2
3-й цикл		4-й цикл		5-й цикл	
Массив чисел	Массив рангов	Массив чисел	Массив рангов	Массив чисел	Массив рангов
0	1	0	1	0	1
2	3	1	4	0	4
0	2	0	2	0	2
1	3	0	3	0	3

В (табл.1) выделен момент, когда одно число из массива обнуляется, при этом оставаясь с тем же рангом, в то время как все остальные ранги увеличиваются на единицу. Процесс сортировки с ранжированием выполнен за пять циклов, поскольку максимальным по значению в массиве является число 5. Размерность массива чисел в данном случае не влияет на продолжительность процесса, поскольку обработка чисел выполняется одновременно над всеми.

Анализируя структуру вычислителя (рис.1) и процесс сортировки (табл.1), время сортировки с ранжированием можно определить по формуле:

$$T_s = (2^q - 1)\tau_{ct} + t_{lg} + \tau_{ct} = 2^q\tau_{ct} + t_{lg}, \quad (1)$$

где  $\tau_{ct}$  – время задержки при счете (операции декремента/инкремента) двоичного счетчика;  $t_{lg}$  – время, затрачиваемое на логические операции в анализаторе реакций;  $q$  – разрядность данных.

Для случая, когда разрядность чисел  $q = 8$ , а размерность массива  $n = 32$ , формула (1) примет вид:

$$T_s = 256\tau_{ct} + t_{lg}, \quad (2)$$

то есть время сортировки в данном случае не зависит от размерности сортируемого массива, а определяется значением максимального числа массива.

В тоже время, учитывая реализацию предложенного способа сортировки с ранжированием на ПЛИС с ориентировочным временем срабатывания базовых узлов в таких микросхемах в наносекундном диапазоне, можно получить конечный результат в миллисекундном диапазоне [8].

**Выводы.** Предложенный способ сортировки с ранжированием использует операции инкремента/декремента для изменения содержимого счетчиков сортировщика и памяти рангов, что позволяет применить их параллельно ко всем элементам числового массива, то есть реализовать вертикальную ассоциативную обработку с формированием рангов без использования операций попарного сравнения чисел с последующей перестановкой.

В функциональном плане такой подход к сортировке чисел за счет формирования рангов позволяет определить не только экстремальные элементы массива чисел, но и элемент, занимающий среднее значение в отсортированном массиве, что характерно для обработки чисел в «окне» апертуры при медианной фильтрации.

#### Список использованных источников:

- [1] Кнут Д.Э. (2003). *Искусство программирования. Т.3. Сортировка и поиск*. Москва: Издательский дом «Вильямс».
- [2] Прэтт У. (1982). *Цифровая обработка изображений*. Москва: Мир.
- [3] Мартинюк Т.Б., Медвідь А.В. & Гуцол О.М. (2013) Моделювання процесу ранжирування значень дискримінантних функцій. *Вісник Вінницького політехнічного інституту*, (5), 74-80.
- [4] Рангайян Р. М. (2007). *Анализ биомедицинских сигналов. Практический подход*. А. П. Немирко (ред.). Москва: Физматлит.
- [5] Юнкеров В.И. & Григорьев С.Г. (2002). *Математико-статистическая обработка данных медицинских исследований*. Санкт-Петербург: ВМедА.
- [6] Лорин Г. (1983). *Сортировка и системы сортировки*. Москва: Мир.
- [7] Кохонен Т. (1982). *Ассоциативные запоминающие устройства*. Москва: Мир.
- [8] Мартинюк Т.Б. & Круківський Б.І. (2020). Модель паралельного сортувальника масиву чисел, *Вісник Вінницького політехнічного інституту*, (5), 49-55.