

Лекция №4**Второй этап реализации сортировки Шелла**

Реализация итераций «чёт-нечётной» перестановки для сформированных на предыдущих этапах под-массивов. Реализуется последовательность действий по обмену между процессами, номера которых соответствуют данному алгоритму и реализация на каждом процессе операции сравнить и разделить. Реализация продолжается, пока содержимое под-массивов изменяется.

Пример реализации итерации алгоритма параллельной сортировки

| Начальный вид | 00 | | | | 01 | | | | 10 | | | | 11 | | | |
|--------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 11 | 50 | 53 | 95 | 36 | 44 | 67 | 86 | 1 | 16 | 35 | 81 | 5 | 15 | 23 | 44 |
| 1-я итерация 1-й этап | 1 | 11 | 16 | 35 | ↓ | | | | 50 | 53 | 81 | 95 | ↓ | | | |
| | ↑ | | | | 5 | 15 | 23 | 36 | ↑ | | | | 44 | 44 | 67 | 86 |
| 2-я итерация 1-й этап | 1 | 5 | 11 | 15 | 16 | 23 | 35 | 36 | ↓ | | | | ↓ | | | |
| | ↑ | | | | ↑ | | | | 44 | 44 | 50 | 53 | 67 | 81 | 86 | 95 |

Каким образом должна быть определена необходимость реализации второго этапа, связанного с «чёт-нечётной» перестановкой?

Пример реализации обмена при $N = 2$ Гиперкуб нарисовать к экзамену

| 1-я итерация | | | 2-я итерация | | | 3-я итерация | | |
|--------------|---|-----|--------------|---|-----|--------------|---|-----|
| 000 | ↔ | 100 | 000 | ↔ | 010 | 000 | ↔ | 001 |
| 001 | ↔ | 101 | 001 | ↔ | 011 | 010 | ↔ | 011 |
| 010 | ↔ | 110 | 100 | ↔ | 110 | 100 | ↔ | 101 |
| 011 | ↔ | 111 | 101 | ↔ | 111 | 110 | ↔ | 111 |

Алгоритм быстрой последовательной сортировки

Метод основывается на последовательном разделении сортируемого массива на блоки меньшего размера таким образом, что бы между элементами разных блоков выполнялось условие упорядочивания. Любой элемент правого блока не меньше любого элемента левого блока.

Особенности реализации

- В левую часть массива помещаются элементы, значение которых не больше значения опорного элемента.
- В правую часть элементы, значение которых больше значения опорного элемента.
- Выбор опорного элемента должен быть такой, что бы исходный массив разделялся на равные части.

Указатели: left – текущий рассматриваемый левый элемент
 right – текущий рассматриваемый правый элемент
 mid – опорный элемент

1. $mid = (mas[left] + mas[right]) / 2$
2. Указатель left смещается на один элемент к концу массива, пока не $mas[left] > mid$
3. Указатель right смещается на один элемент к началу, пока не $mas[right] \leq mid$
4. Действия связанные с проверкой рассматриваемых условий и обменом элементов, находящихся в соответствующих позициях (left и right) выполняются до тех пор, пока значения этих указателей не станут равными

Пример реализации алгоритма

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 7 | 2 | 5 | 9 | 1 | 3 | 8 |
| min = 1; max = 9; mid=5; left = 1; right = 8; | | | | | | | |
| mas[1] > mid => left = 1; mas[8] < mid => right = 7; | | | | | | | |
| Выполняется обмен | | | | | | | |
| 3 | 7 | 2 | 5 | 9 | 1 | 6 | 8 |
| mas[1] < mid => left = 2; mas[1] < mid => left = 2; mas[1] < mid => left = 2; | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |