Pigeonhole sorting

PARALLEL PROGRAMMING

-PASUPULETI ROHITH SAI DATTA

-CHINMAYA DAYANANDA KAMATH

Introduction

- Pigeonhole sorting works well when the number of elements is close to the number of key values.
- Pigeonhole sorting takes O(n + Range) time, where 'n' is the number of elements, and 'Range' is the possible value range in the array.
- Pigeonhole sorting is a non-comparison-based sort, which makes it faster for certain applications.
- Pigeonhole sorting is a stable sorting algorithm.
- Pigeonhole sorting operates with a linear time complexity for sorting.

Working of Algorithm

- 1. Find minimum and maximum values in the array. Let the minimum and maximum values be ‘min’ and ‘max’ respectively. Also, find the range as ‘max-min+1’.
- 2. Set up an array of initially empty “pigeonholes” the same size as the range.
- 3. Visit each array element and then put each element in its pigeonhole. An element arr[i] is put in the hole at index arr[i] – min.]
- 4. Start the loop all over the pigeonhole array in order and put the elements from non-empty holes back into the original array.

Sequential pseudo code

```
void pigeonhole_sort(int arr[], int n, int min_val, int range_size, int* sorted_arr) {
    int pigeonholes[range_size];
    for (int i = 0; i < range_size; i++) {
        pigeonholes[i] = 0;
    }

    for (int i = 0; i < n; i++) {
        pigeonholes[arr[i] - min_val]++;
    }

    int index = 0;
    for (int i = 0; i < range_size; i++) {
        while (pigeonholes[i] > 0) {
            sorted_arr[index++] = i + min_val;
            pigeonholes[i]--;
        }
    }
}
```

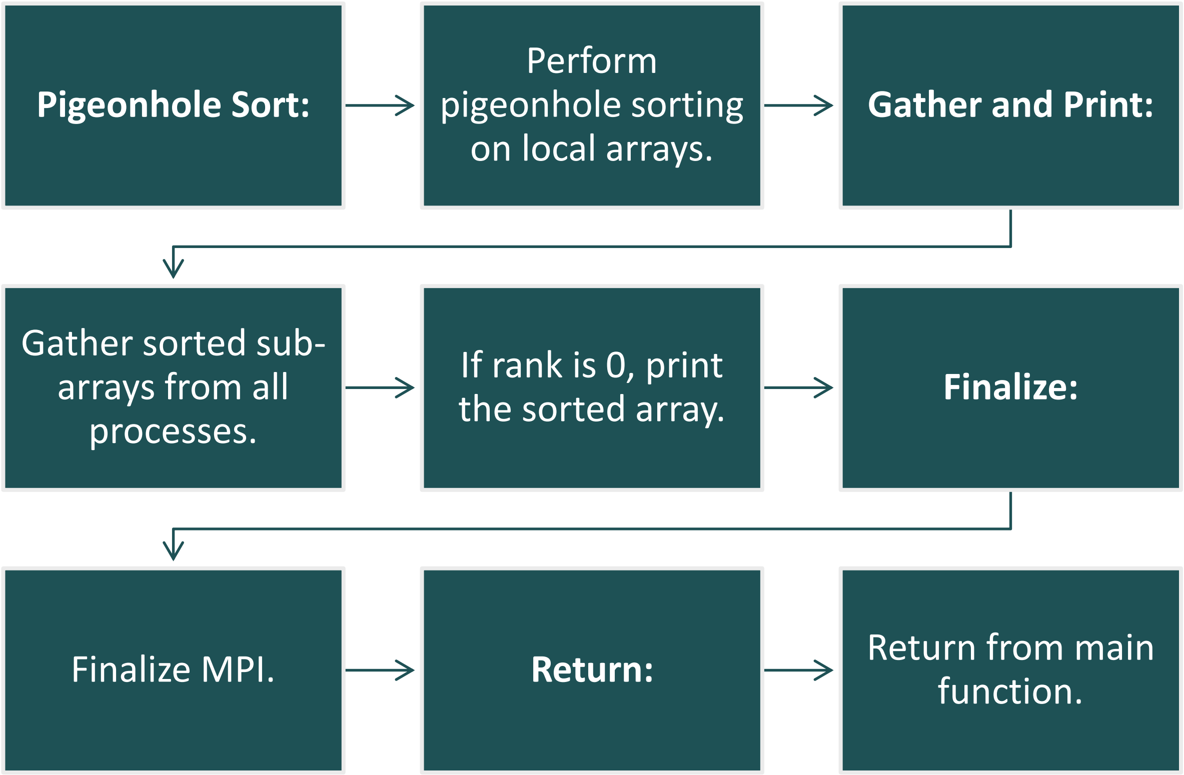
parallel algorithm idea

- **Initialize:**
- Import necessary libraries.
- Define the pigeonhole sort function.
- Main Function:

- **Initialize MPI:**
- Get process rank and size.
- If rank is 0, input total number of elements.
- Broadcast total elements to all processes.
- Divide data among processes using Scatter.

- **Local Sorting:**
- Find local minimum and maximum values.
- Reduce to get global minimum and maximum values.
- Calculate range size for pigeonhole sorting.

Parallel algorithm idea



Pigeonhole Sort:

Perform pigeonhole sorting on local arrays.

Gather and Print:

Gather sorted sub-arrays from all processes.

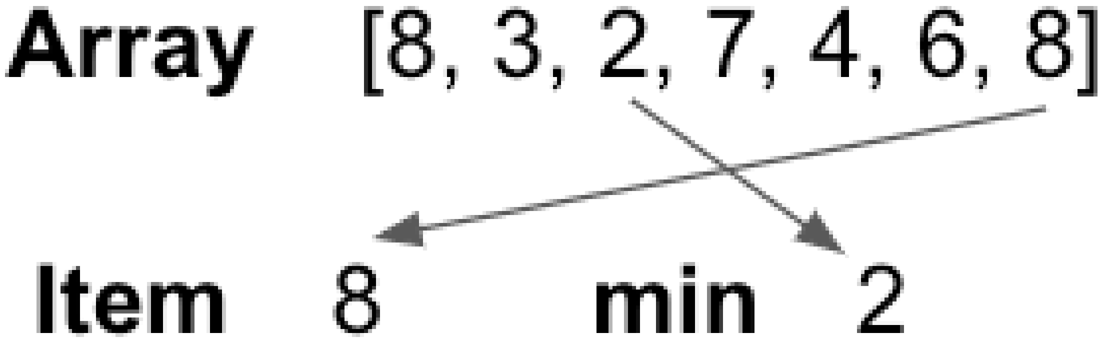
If rank is 0, print the sorted array.

Finalize:

Finalize MPI.

Return:

Return from main function.



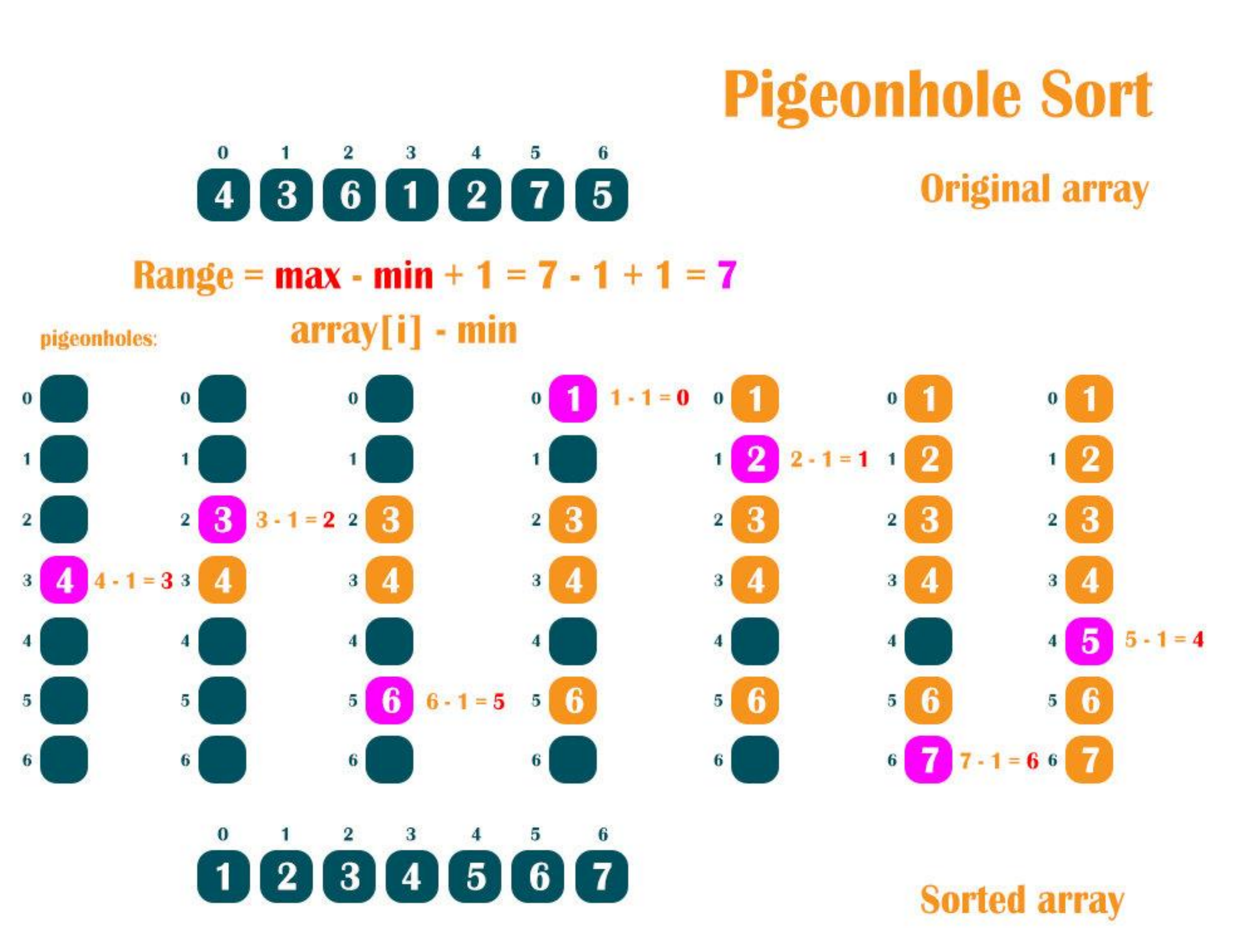
Array [8, 3, 2, 7, 4, 6, 8]

Item 8 **min** 2

Item - min = 8 - 2 = 6

Holes

0	2
1	3
2	4
3	
4	6
5	7
6	8, 8



Pigeonhole Sort

Original array

$$\text{Range} = \max - \min + 1 = 7 - 1 + 1 = 7$$

array[i] - min

pigeonholes:

Sorted array

Thank You