Binary Search using Recursion!

Given a sorted array, return the index of an element x in the array. Use binary search to find the element in the array !
Return -1 if the element is not present.

Input Format

In the function an integer vector is passed.

Output Format

Return an integer denoting index of the element to be find.

Sample Input

```
{1, 3, 5, 7, 9}, x = 3
```

Sample Output

```
1
```

Solution: recBinarySearch.cpp

2D Array Merge

Implement merge sort for a two-dimensional array. In case of odd dimension, the first division contains more number of elements than the second one. The complete execution of merge sort arranges the elements in increasing order either moving row-wise or column-wise.

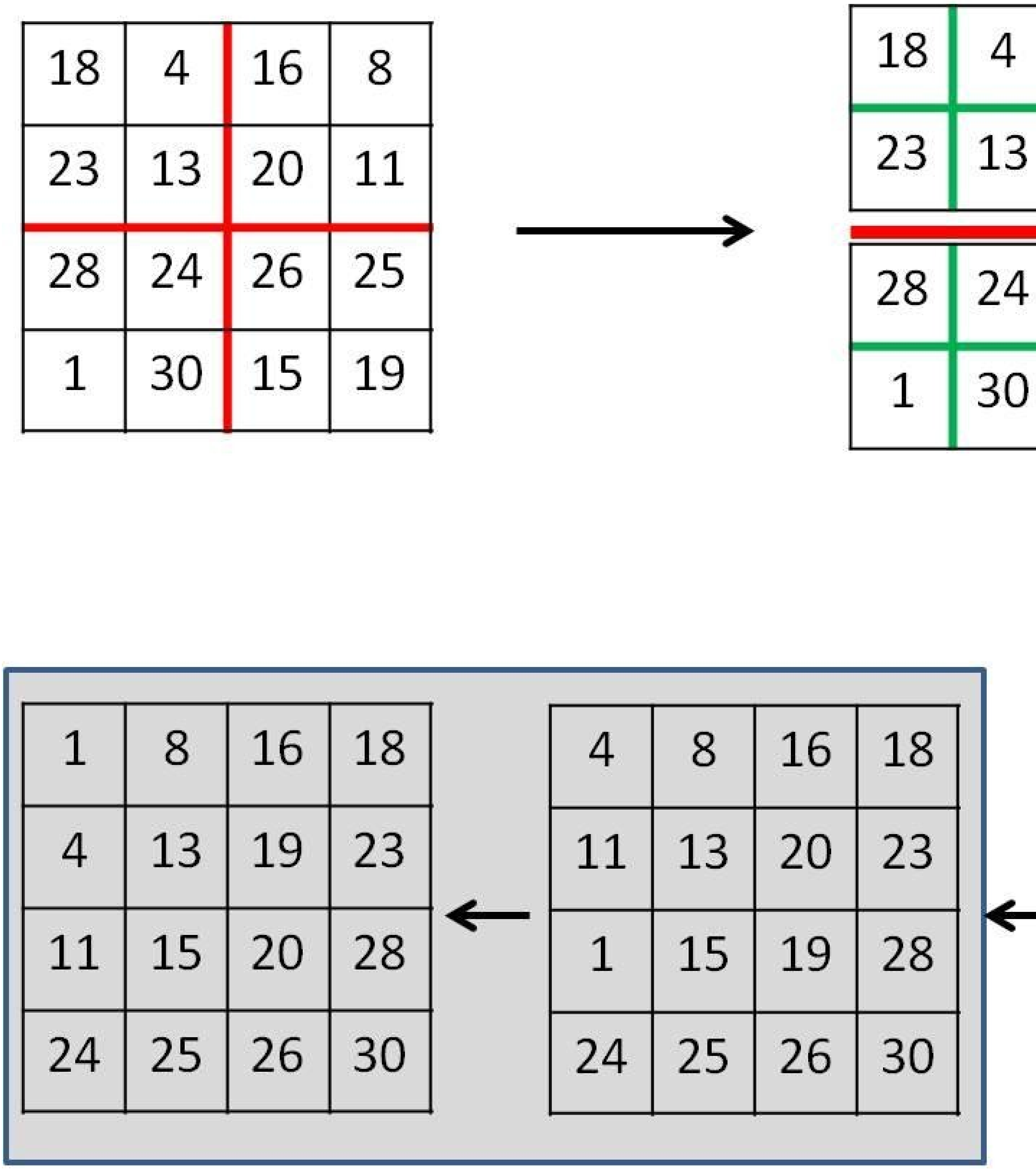
For example, let there be a

4×4

4×4 two dimensional array. The complete process to be implemented is illustrated in Fig. 1. Similarly, Fig. 2 demonstrates the scenario for a

3×3

3×3 two dimensional array. One has to keep on dividing till a single element is remaining. During merging, first the row elements get sorted in increasing order followed by sorting of elements lying in the same column.

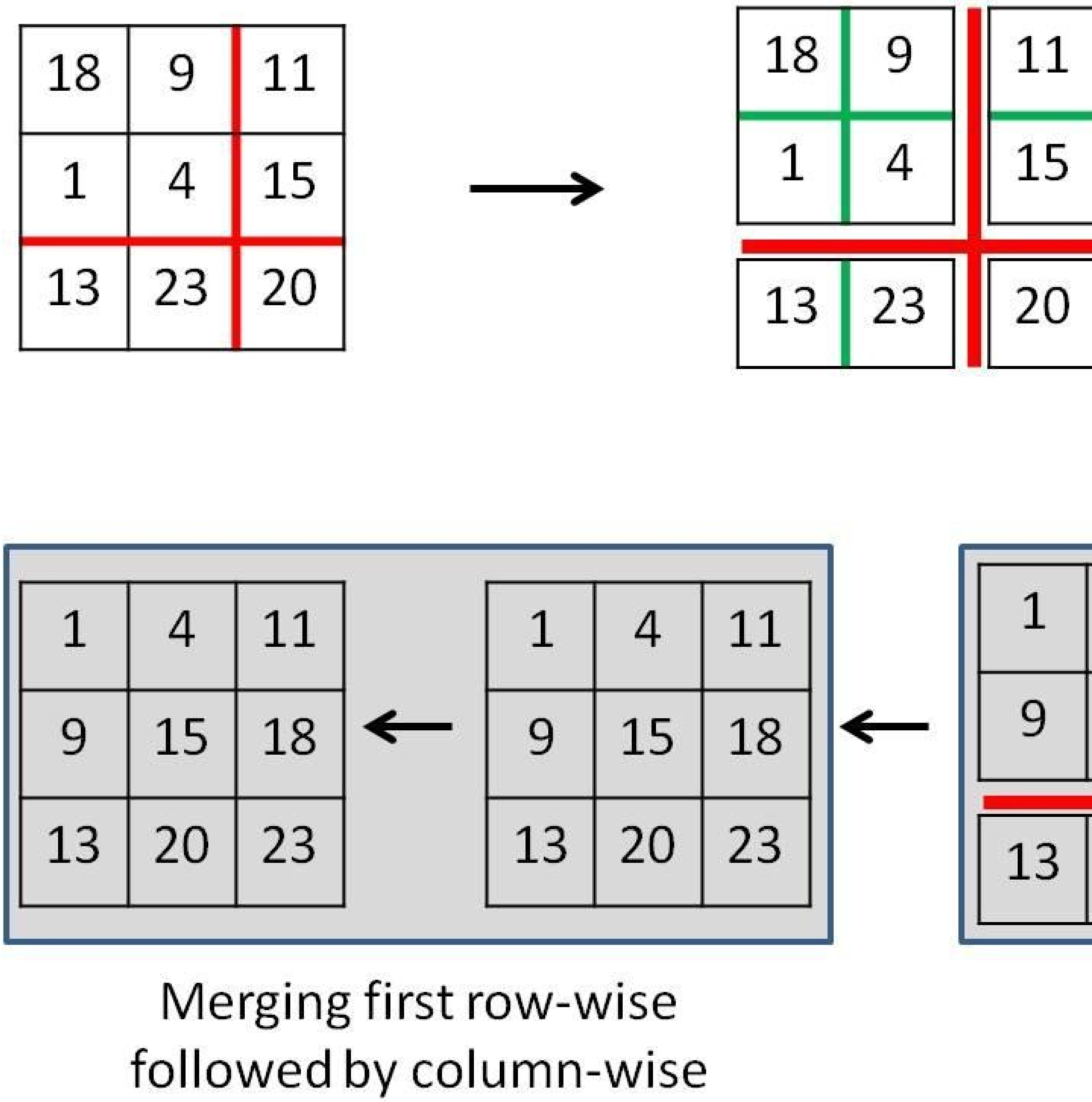


Merging first row-wise
followed by column-wise

Fig.1 - 2D Mergesort on a

4×4

4×4 two dimensional array.



Solution: mergeSort.cpp