Merge Sort

Prof. Pramod Nath
A.P, KIET

What is Merge Sort?

Merge sort is a **sorting** technique based on divide and conquer technique. With worst-case time complexity being O(n log n), it is one of the most respected algorithms.

Merge sort first divides the array into equal halves and then combines them in a **sorted** manner.

Definition:

- Merge sort is a **DIVIDE AND CONQUER** algorithm.

It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves.

The merge( ) function is used for merging two halves.

Steps involved:

– **Divide the problem into sub-problems that** are similar to the original but smaller in size.

– **Conquer the sub-problems by solving them** recursively. If they are small enough, just solve them in a straightforward manner.

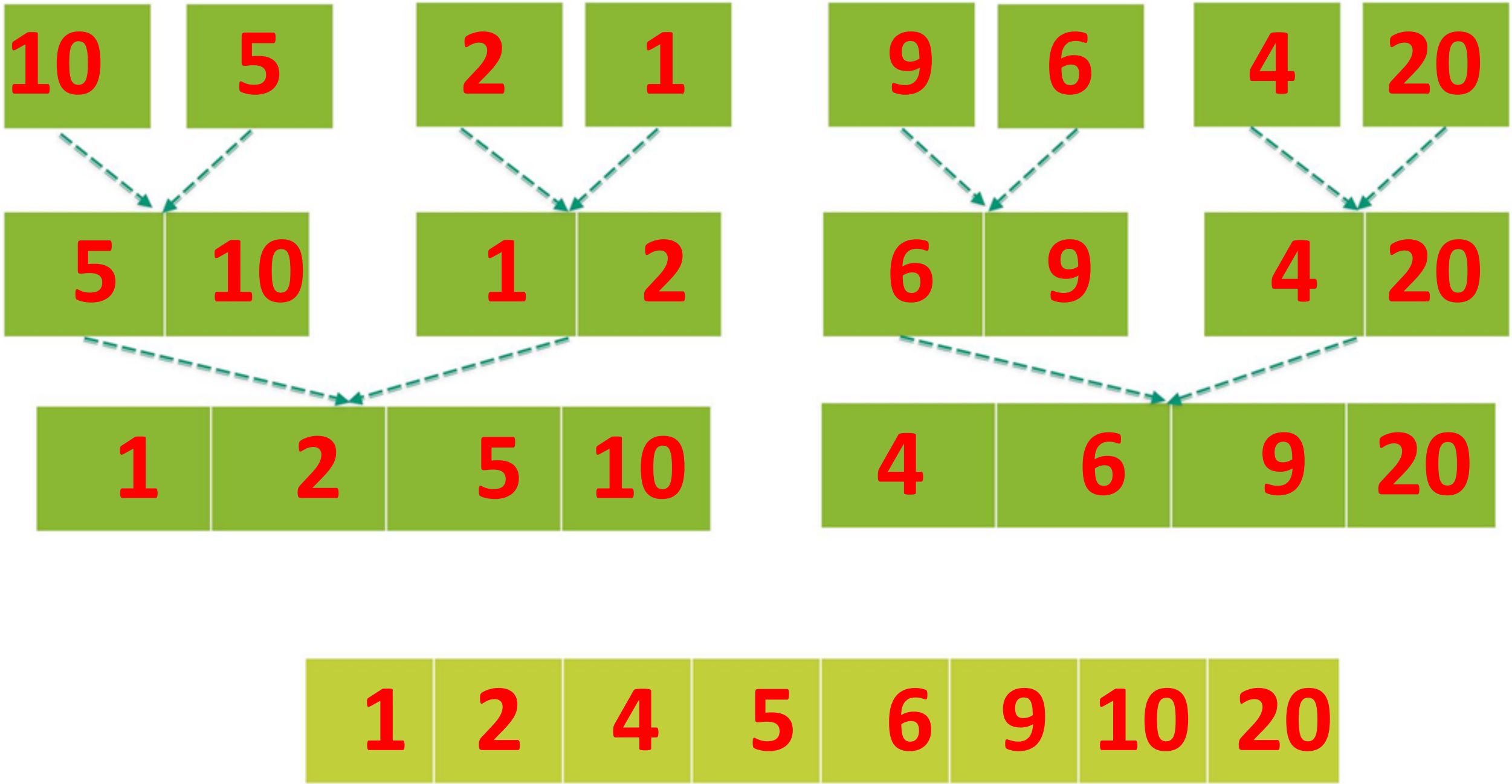
– **Combine the solutions to create a solution to** the original problem.

Why Merge Sort??

- Compared to insertion sort merge sort is faster.
- On small inputs, insertion sort may be faster.

But for large enough inputs, merge sort will always be faster, because its running time grows more slowly than insertion sorts.

Merge Sort Example


10
5
2
1
9
6
4
20
5
10
1
2
6
9
4
20
1
2
5
10
4
6
9
20
1
2
4
5
6
9
10
20

Main part of Code:

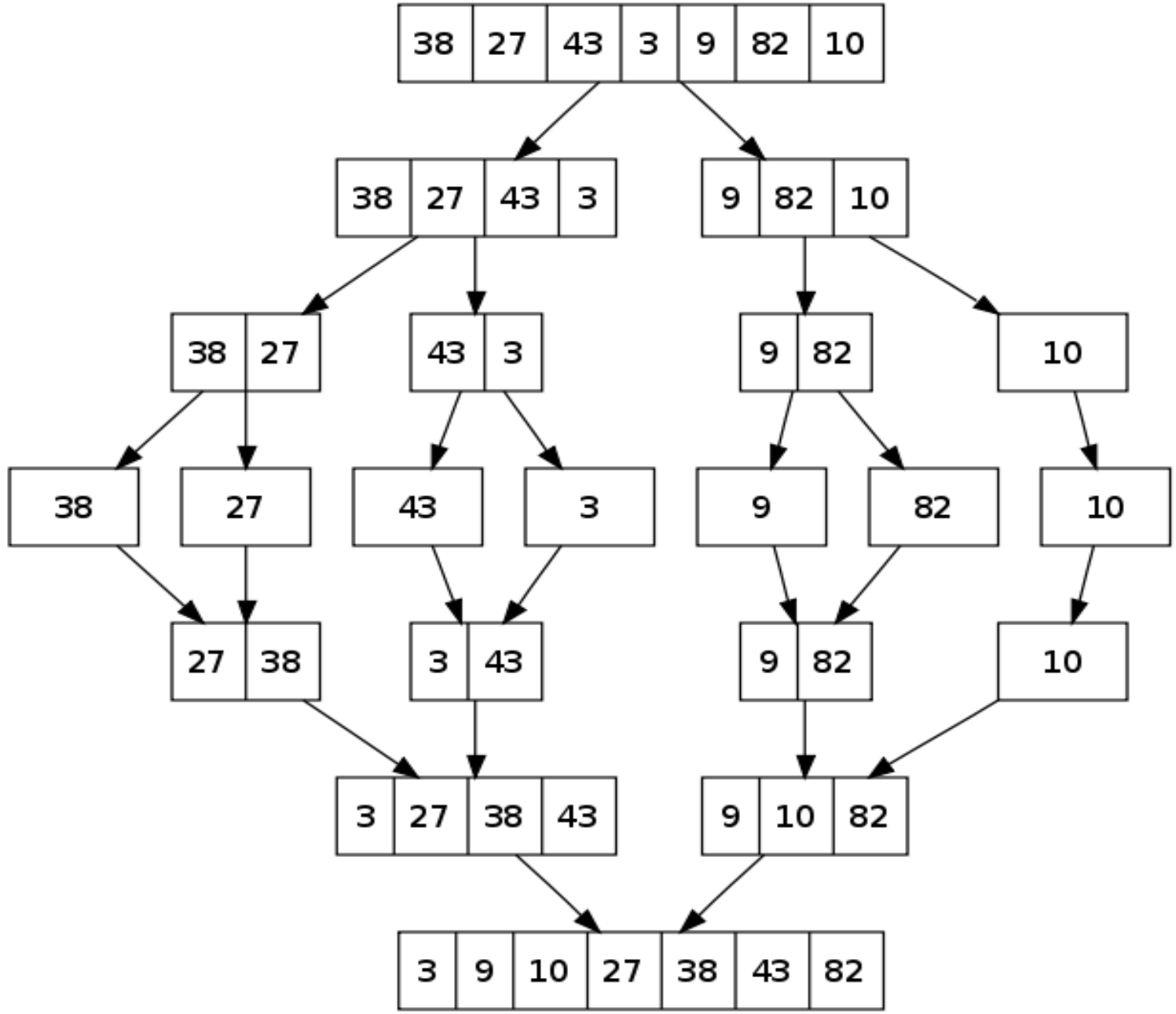
```
mergesort (int a[ ], int low, int high)
{
int mid;
if ( low < high)
{
mid = (low + high) / 2;
mergesort(a, low, mid);
mergesort(a, mid+1, high);
merge(a, low, high);
}
}
```

```
void merge ( int a[ ], int low, int mid, int high)
{
        int i, j, k;
        int n1 = mid – low + 1;
        int n2 = high – mid ;
        int L[n1], R[n2] ;
        for( i= 0; i < n1 ; i++)
                L[i] = a[low + i];
        for( j= 0; j < n2 ; j++)
                R[j] = a[mid + 1 + j];
```

```
 i = 0;          j = 0;          k = low;
while( i < n1 && j < n2 )
{
        if( L[i] <= R[j] )
        {
                a[k] = L[i];
                i++;
        }
        else{
                a[k] = R[j];
                j++;
        }
        k++;
}/* while loop closed */
```

```
 while( i < n1)
{
        a[k] = L[i];
        i++;
        k++;
}

 while( j < n2)
{
        a[k] = R[j];
        j++;
        k++;
}
}/* merge( ) closed */
```


38 27 43 3 9 82 10
38 27 43 3
9 82 10
38 27
43 3
9 82
10
38
27
43
3
9
82
10
27 38
3 43
9 82
10
3 27 38 43
9 10 82
3 9 10 27 38 43 82

About Merge Sort

>Merge sort follows recursive algorithom……

>We divide the array into halves till the sub array has only 1 element.

>Major work is done in merging the sub arrays……

>Merge Sort is a Stable Sort.

>We need an extra temporary array of the same size as the input array for merging. So, Merge Sort is an example of Out-Place Sorting.

Any questions?

THANK YOU