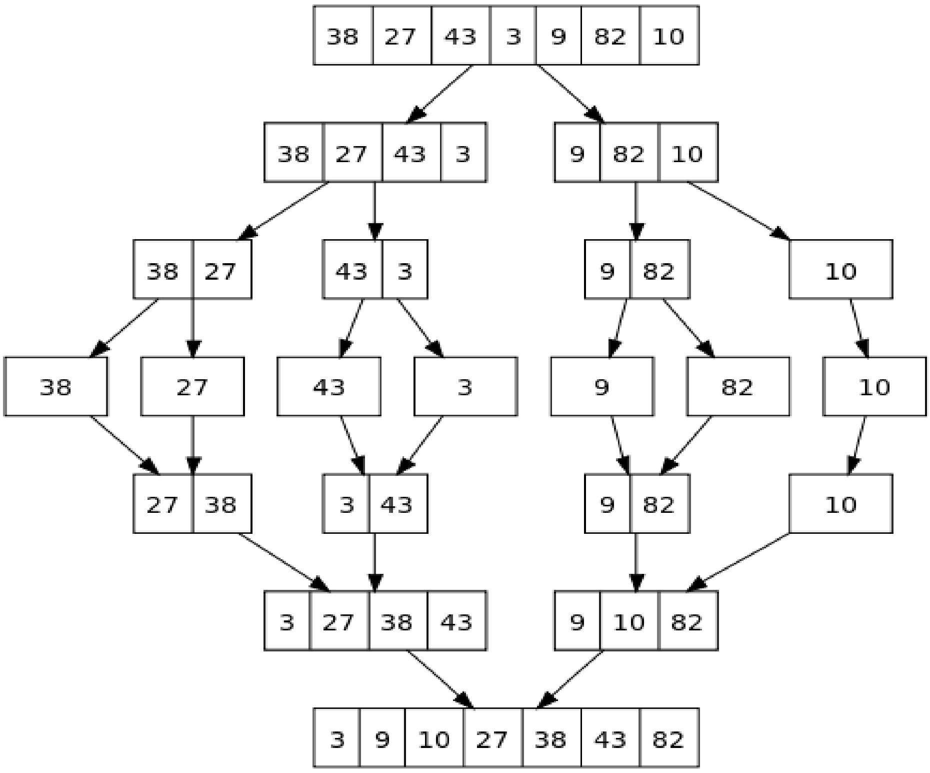
4. Merge Sort:

Merge Sort follows the **Divide and Conquer** mechanism to sort a given set of numbers recursively, hence consuming less time compared to other sorting algorithms.

The concept of Divide and Conquer involves three steps:

1. **Divide:** Divide the problem into multiple small problems.
2. **Conquer:** Conquer the subproblems by solving them.
3. **Combine:** Combine the solutions of the subproblems to find the solution of the actual problem.

Example:



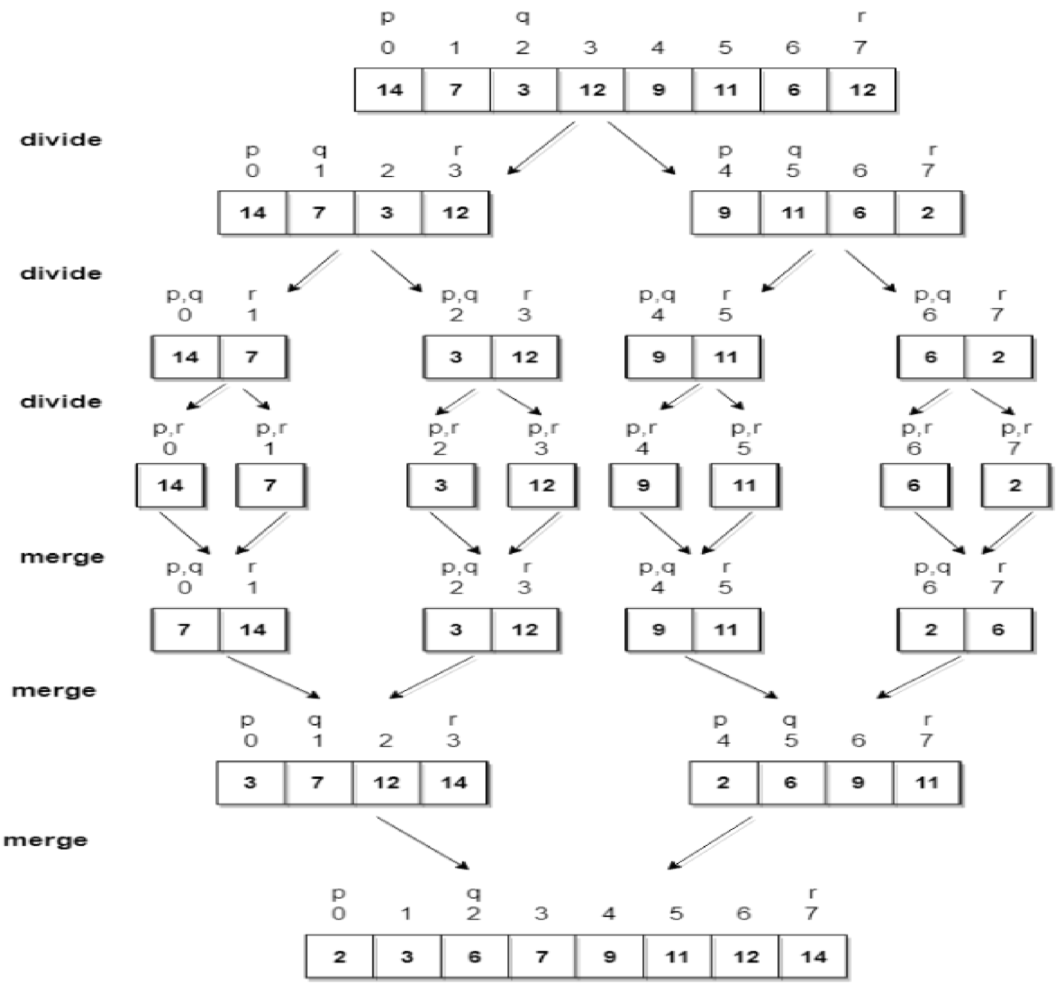
Algorithm:

// merge sort function

```
void mergeSort(int a[], int p, int r)
{
    int q;
    if(p<r)
     {
        q=(p+r)/2;
        mergeSort(a,p,q);
        mergeSort(a,q+1,r);
        merge(a,p,q,r); //function to merge the subarrays
    }
}
```

```
// function to merge the subarrays
void merge(int a[], int p, int q, int r)
     {
     int b[5];   //same size of a[]
     int i,j,k;
     k=0;
     i=p;
     j=q + 1;
     while(i<=q && j<=r)
          {
          if(a[i]<a[j])
               {
               b[k]=a[i];
               k++;
               i++;
               }
          else
               {
               b[k]=a[j];
               k++;
               j++;
               }
          }
     while(i <= q)
          {
          b[k]=a[i];
          k++;
          i++;
          }
     while(j <= r)
          {
          b[k]=a[j];
          k++;
          j++;
          }
  // copying back the sorted list from b[] to a[]
    for(i=p;i<=r;i++)
          {
        a[i]=b[i];
          }
     } //end of merge function
```

Example:


p q r
0 1 2 3 4 5 6 7
14 7 3 12 9 11 6 12
divide
p q r p q r
0 1 2 3 4 5 6 7
14 7 3 12 9 11 6 2
divide
p,q r p,q r p,q r p,q r
0 1 2 3 4 5 6 7
14 7 3 12 9 11 6 2
divide
p,r p,r p,r p,r p,r p,r p,r p,r
0 1 2 3 4 5 6 7
14 7 3 12 9 11 6 2
merge
p,q r p,q r p,q r p,q r
0 1 2 3 4 5 6 7
7 14 3 12 9 11 2 6
merge
p q r p q r
0 1 2 3 4 5 6 7
3 7 12 14 2 6 9 11
merge
p q r
0 1 2 3 4 5 6 7
2 3 6 7 9 11 12 14