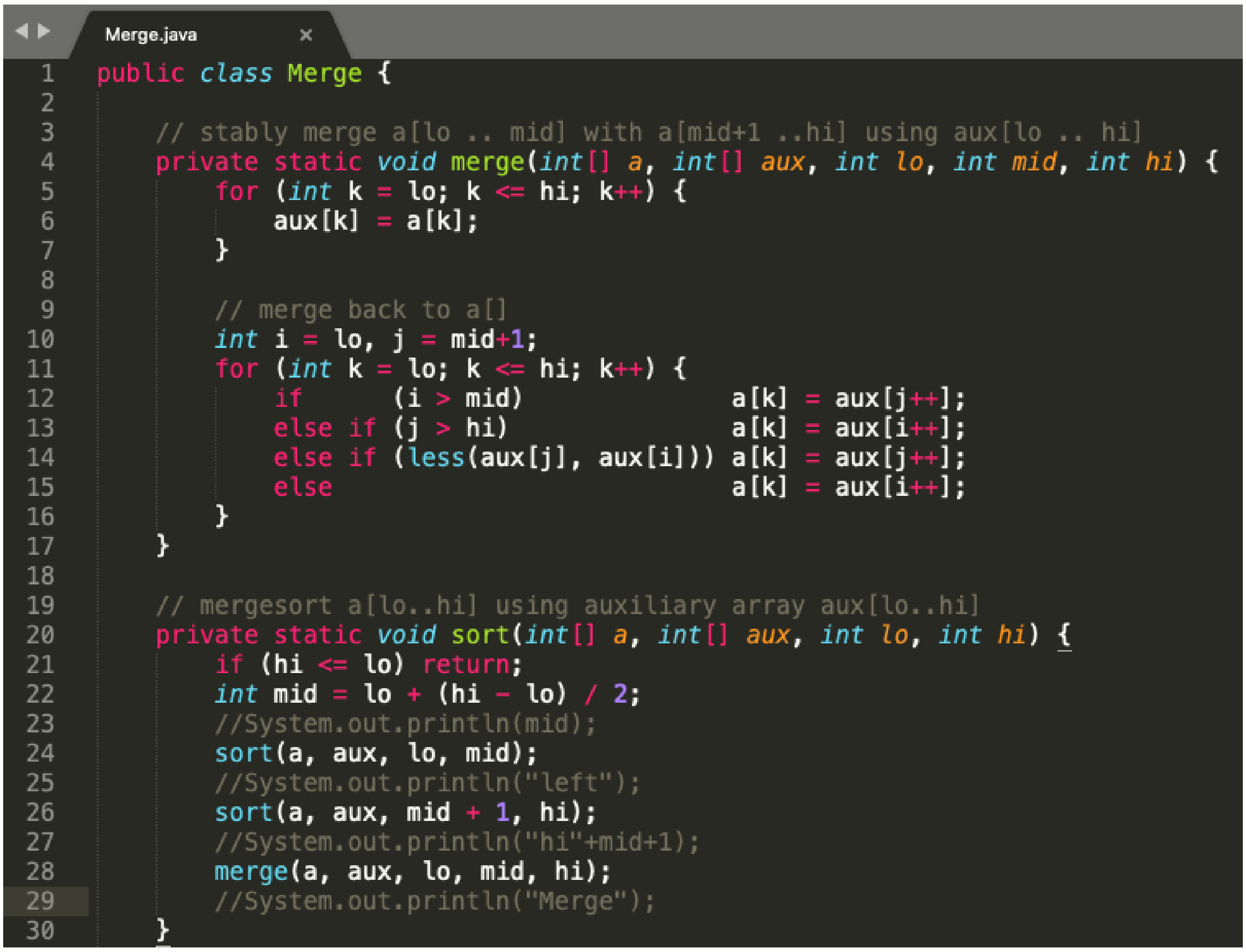
ROYAL UNIVERSITY OF BHUATAN
GYALPOZHING COLLEGE OF INFORMATION TECHNOLOGY
GYALPOZHING : BHUTAN

MERGESORT

Idea:

1. Divide the unsorted list into N sublists, each containing 1 element.
2. Take adjacent pairs of two singleton lists and merge them to form a list of 2 elements. N will now convert into N/2 lists of size 2.
3. Repeat the process till a single sorted list of obtained.

Program of MergeSort in JAVA



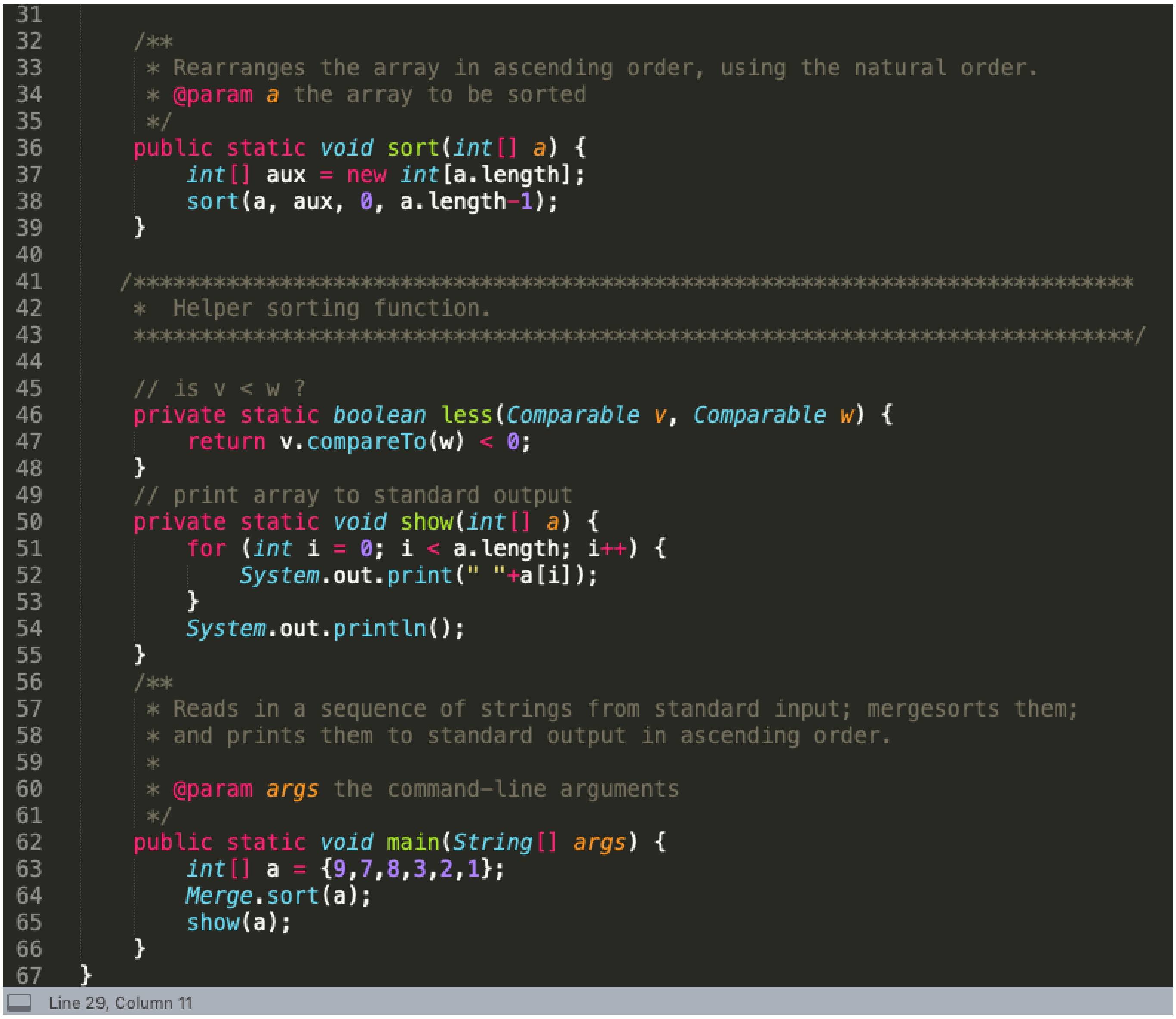
```java
public class Merge {

    // stably merge a[lo .. mid] with a[mid+1 ..hi] using aux[lo .. hi]
    private static void merge(int[] a, int[] aux, int lo, int mid, int hi) {
        for (int k = lo; k <= hi; k++) {
            aux[k] = a[k];
        }

        // merge back to a[]
        int i = lo, j = mid+1;
        for (int k = lo; k <= hi; k++) {
            if      (i > mid)              a[k] = aux[j++];
            else if (j > hi)               a[k] = aux[i++];
            else if (less(aux[j], aux[i])) a[k] = aux[j++];
            else                           a[k] = aux[i++];
        }
    }

    // mergesort a[lo..hi] using auxiliary array aux[lo..hi]
    private static void sort(int[] a, int[] aux, int lo, int hi) {
        if (hi <= lo) return;
        int mid = lo + (hi - lo) / 2;
        //System.out.println(mid);
        sort(a, aux, lo, mid);
        //System.out.println("left");
        sort(a, aux, mid + 1, hi);
        //System.out.println("hi"+mid+1);
        merge(a, aux, lo, mid, hi);
        //System.out.println("Merge");
    }
```

Figure 1: code 1



```java

    /**
     * Rearranges the array in ascending order, using the natural order.
     * @param a the array to be sorted
     */
    public static void sort(int[] a) {
        int[] aux = new int[a.length];
        sort(a, aux, 0, a.length-1);
    }

   /***************************************************************************
    *  Helper sorting function.
    ***************************************************************************/

    // is v < w ?
    private static boolean less(Comparable v, Comparable w) {
        return v.compareTo(w) < 0;
    }
    // print array to standard output
    private static void show(int[] a) {
        for (int i = 0; i < a.length; i++) {
            System.out.print(" "+a[i]);
        }
        System.out.println();
    }
    /**
     * Reads in a sequence of strings from standard input; mergesorts them;
     * and prints them to standard output in ascending order.
     *
     * @param args the command-line arguments
     */
    public static void main(String[] args) {
        int[] a = {9,7,8,3,2,1};
        Merge.sort(a);
        show(a);
    }
}
```

Line 29, Column 11

Figure 2: continue code 1

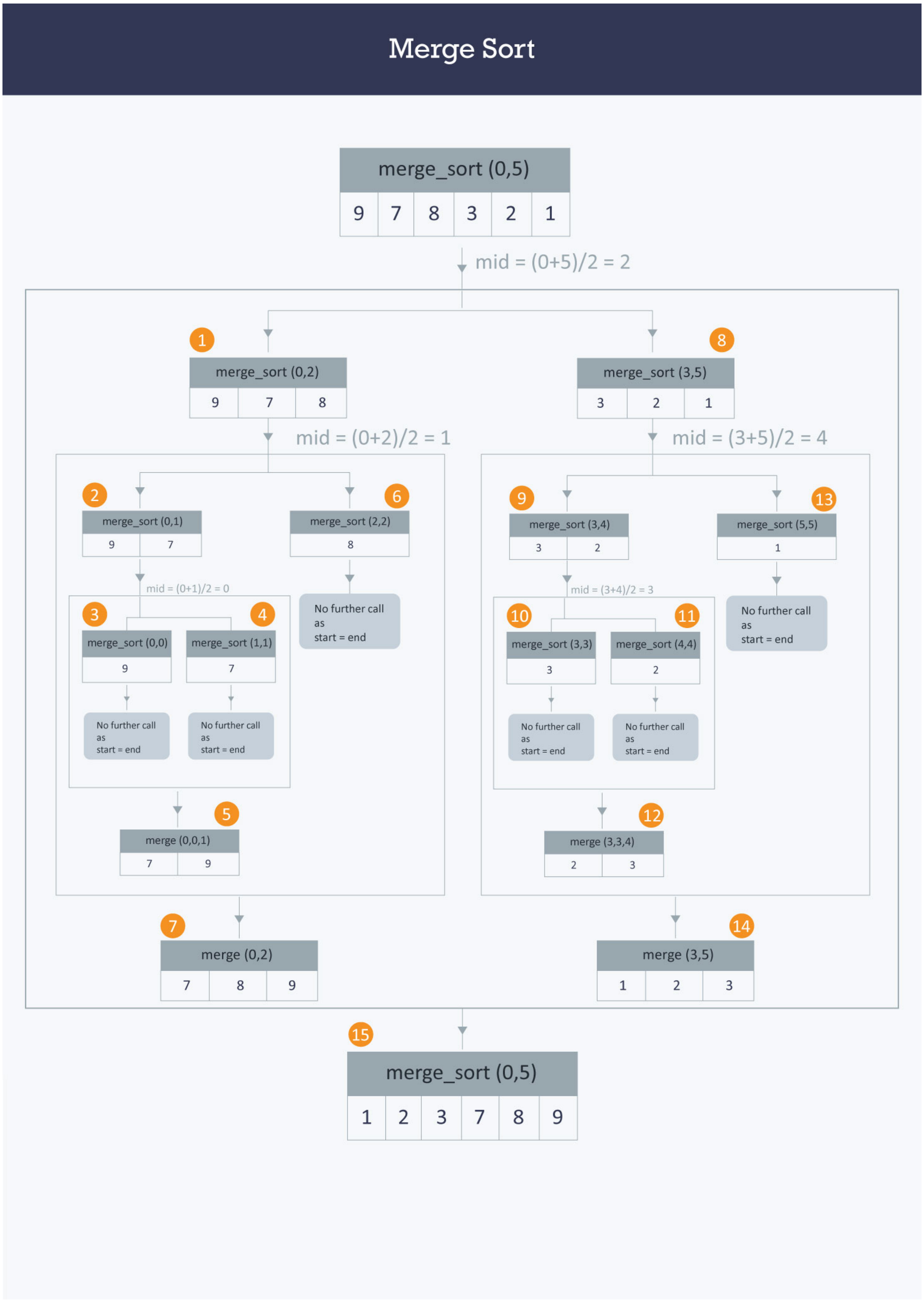


Figure 3: Visualization