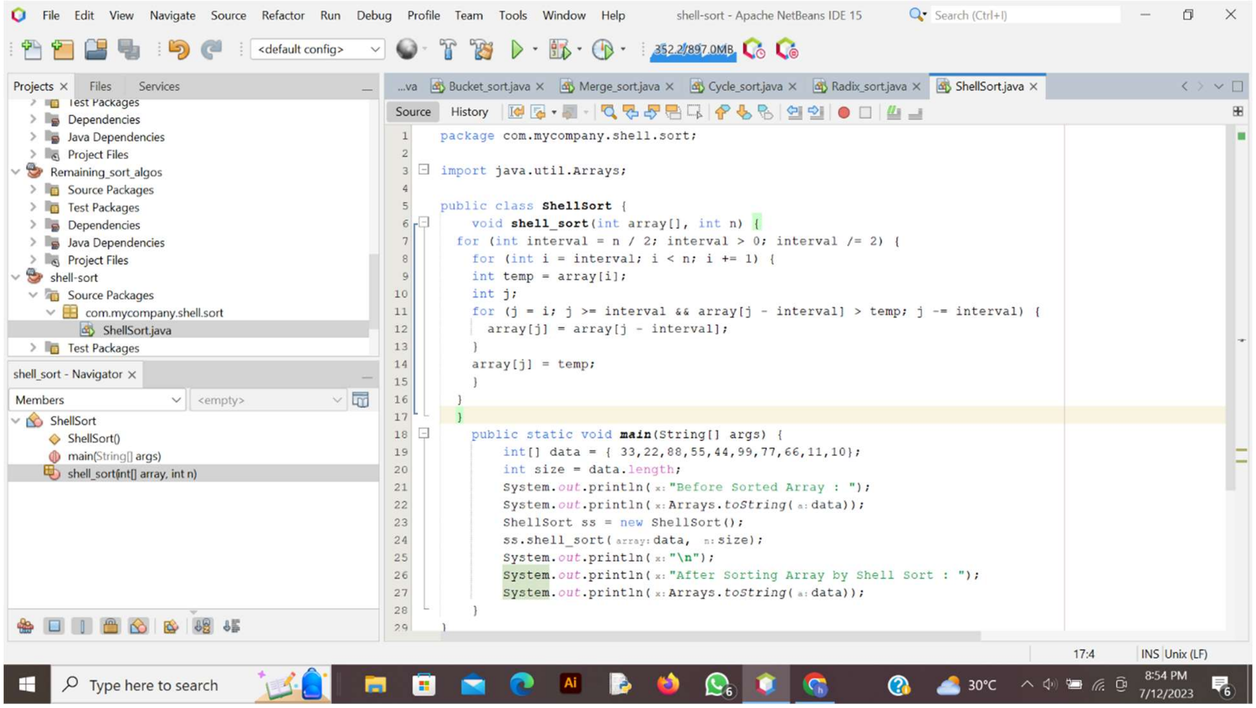
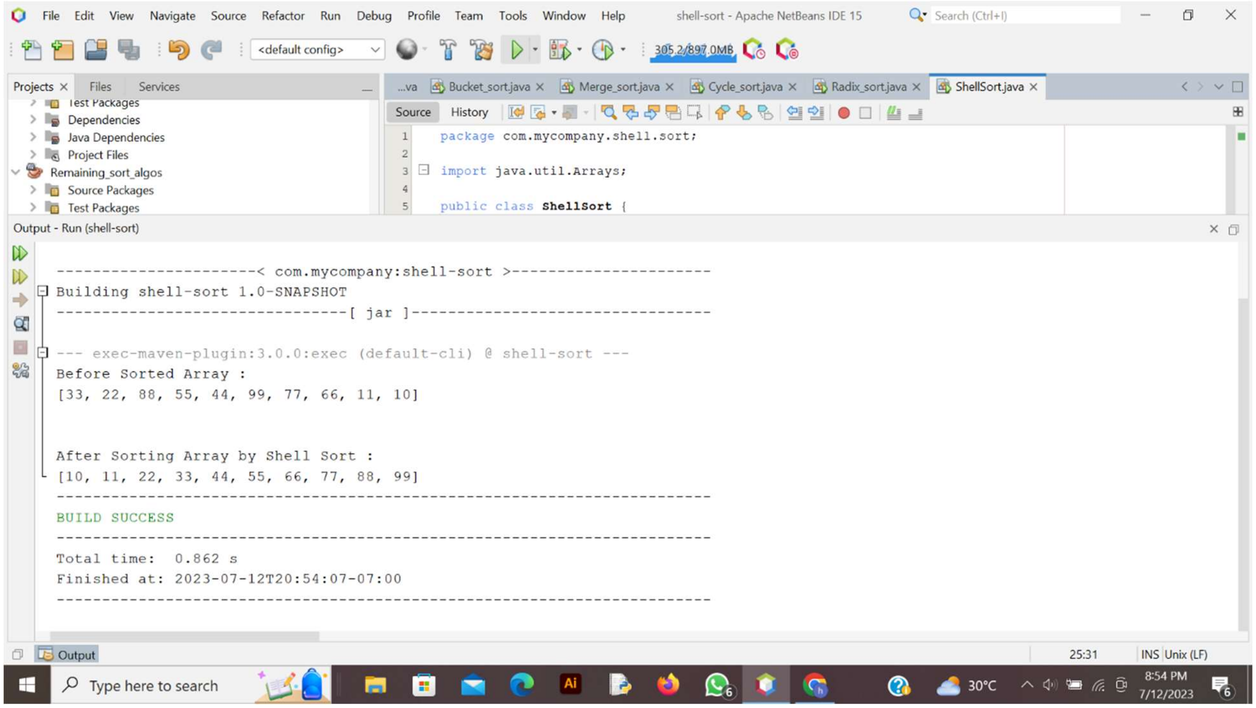
Today’s Task (Sorting Algorithm)

Shell Sort



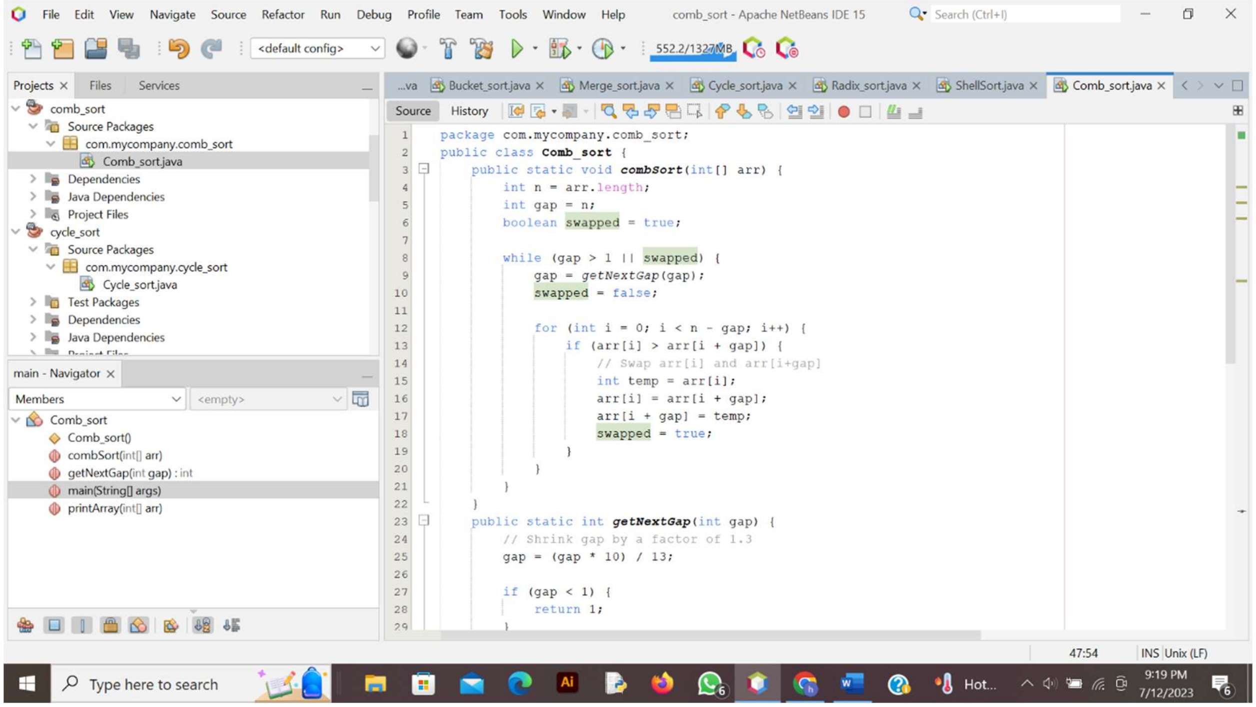
```java
package com.mycompany.shell.sort;

import java.util.Arrays;

public class ShellSort {
    void shell_sort(int array[], int n) {
  for (int interval = n / 2; interval > 0; interval /= 2) {
    for (int i = interval; i < n; i += 1) {
    int temp = array[i];
    int j;
    for (j = i; j >= interval && array[j - interval] > temp; j -= interval) {
      array[j] = array[j - interval];
    }
    array[j] = temp;
    }
  }
  }
    public static void main(String[] args) {
        int[] data = { 33,22,88,55,44,99,77,66,11,10};
        int size = data.length;
        System.out.println( x: "Before Sorted Array : ");
        System.out.println( x: Arrays.toString( a: data));
        ShellSort ss = new ShellSort();
        ss.shell_sort( array: data,  n: size);
        System.out.println( x: "\n");
        System.out.println( x: "After Sorting Array by Shell Sort : ");
        System.out.println( x: Arrays.toString( a: data));
    }
}
```



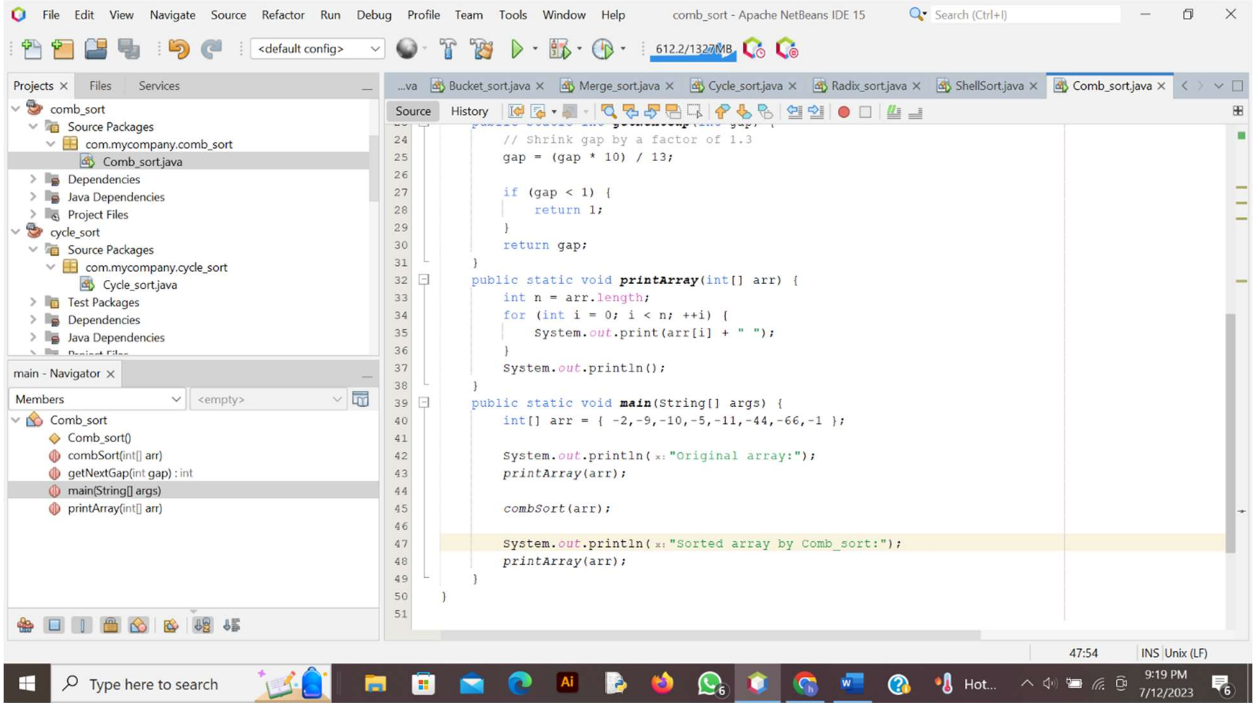
```java
package com.mycompany.shell.sort;

import java.util.Arrays;

public class ShellSort {
```

```
---------------------< com.mycompany:shell-sort >----------------------
Building shell-sort 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ shell-sort ---
Before Sorted Array :
[33, 22, 88, 55, 44, 99, 77, 66, 11, 10]


After Sorting Array by Shell Sort :
[10, 11, 22, 33, 44, 55, 66, 77, 88, 99]
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.862 s
Finished at: 2023-07-12T20:54:07-07:00
------------------------------------------------------------------------
```

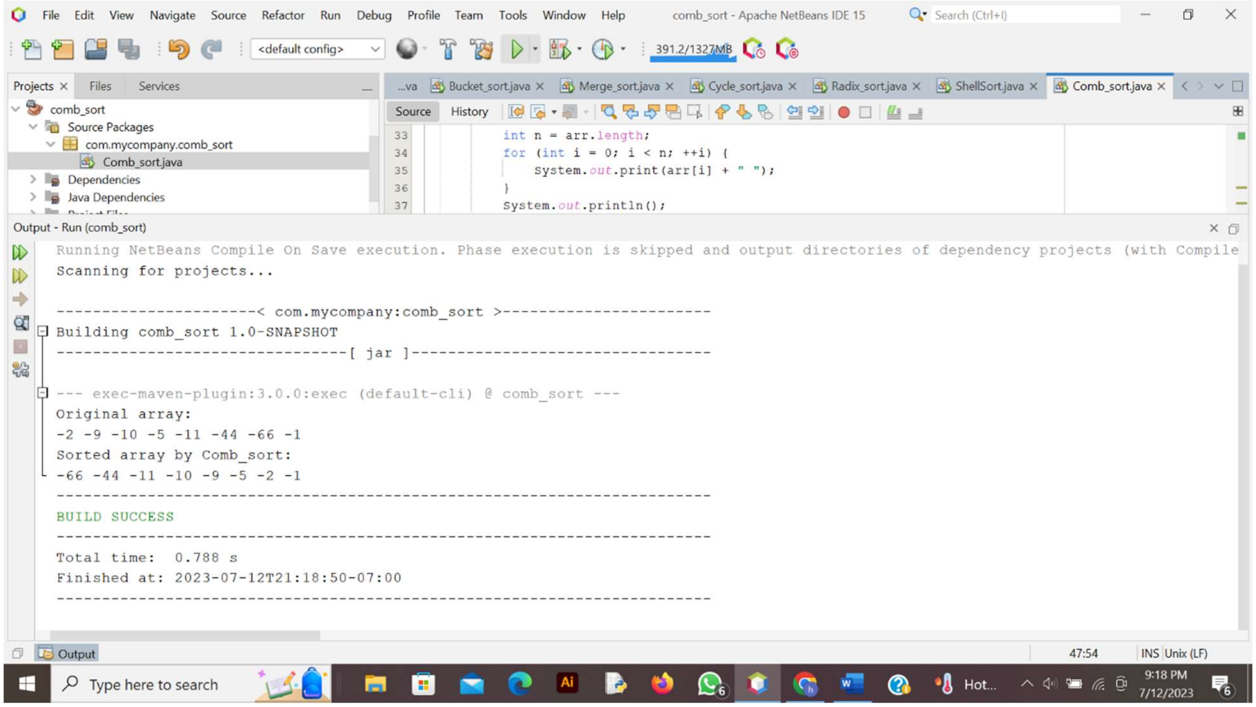
Comb Sort



```java
package com.mycompany.comb_sort;
public class Comb_sort {
    public static void combSort(int[] arr) {
        int n = arr.length;
        int gap = n;
        boolean swapped = true;

        while (gap > 1 || swapped) {
            gap = getNextGap(gap);
            swapped = false;

            for (int i = 0; i < n - gap; i++) {
                if (arr[i] > arr[i + gap]) {
                    // Swap arr[i] and arr[i+gap]
                    int temp = arr[i];
                    arr[i] = arr[i + gap];
                    arr[i + gap] = temp;
                    swapped = true;
                }
            }
        }
    }
    public static int getNextGap(int gap) {
        // Shrink gap by a factor of 1.3
        gap = (gap * 10) / 13;

        if (gap < 1) {
            return 1;
        }
```



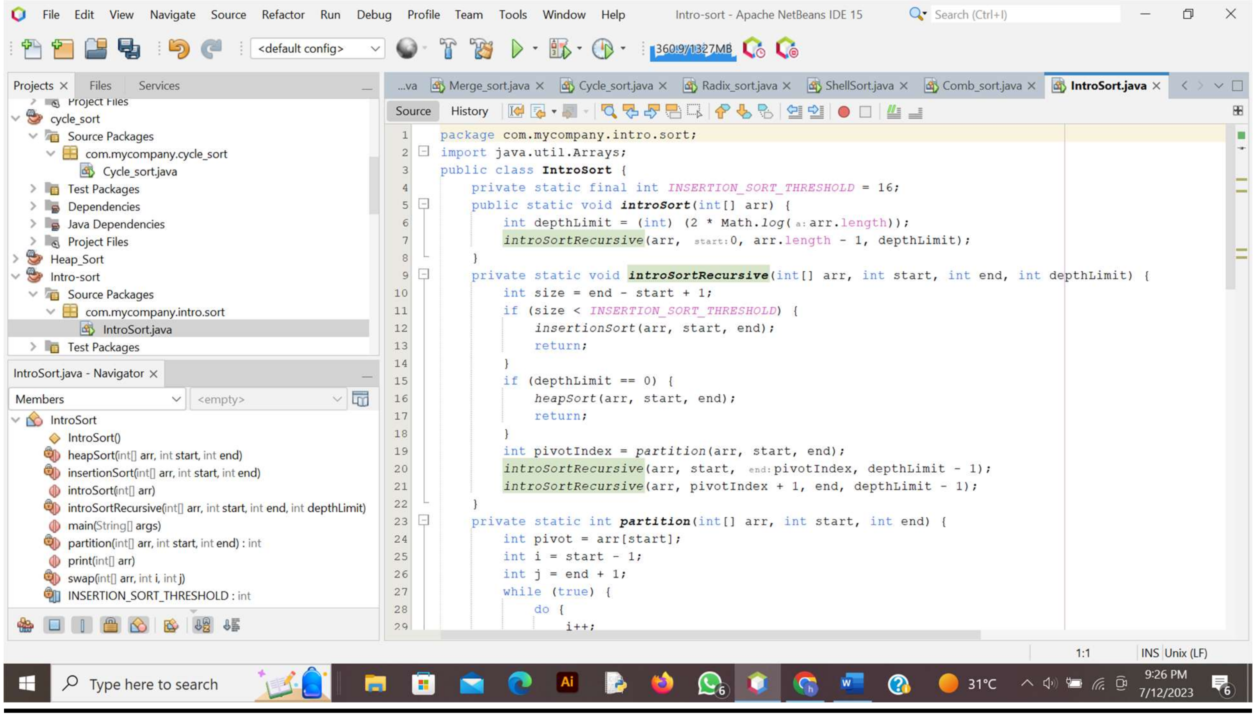
```java
        // Shrink gap by a factor of 1.3
        gap = (gap * 10) / 13;

        if (gap < 1) {
            return 1;
        }
        return gap;
    }
    public static void printArray(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n; ++i) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] arr = { -2,-9,-10,-5,-11,-44,-66,-1 };

        System.out.println( x: "Original array:");
        printArray(arr);

        combSort(arr);

        System.out.println( x: "Sorted array by Comb_sort:");
        printArray(arr);
    }
}

```



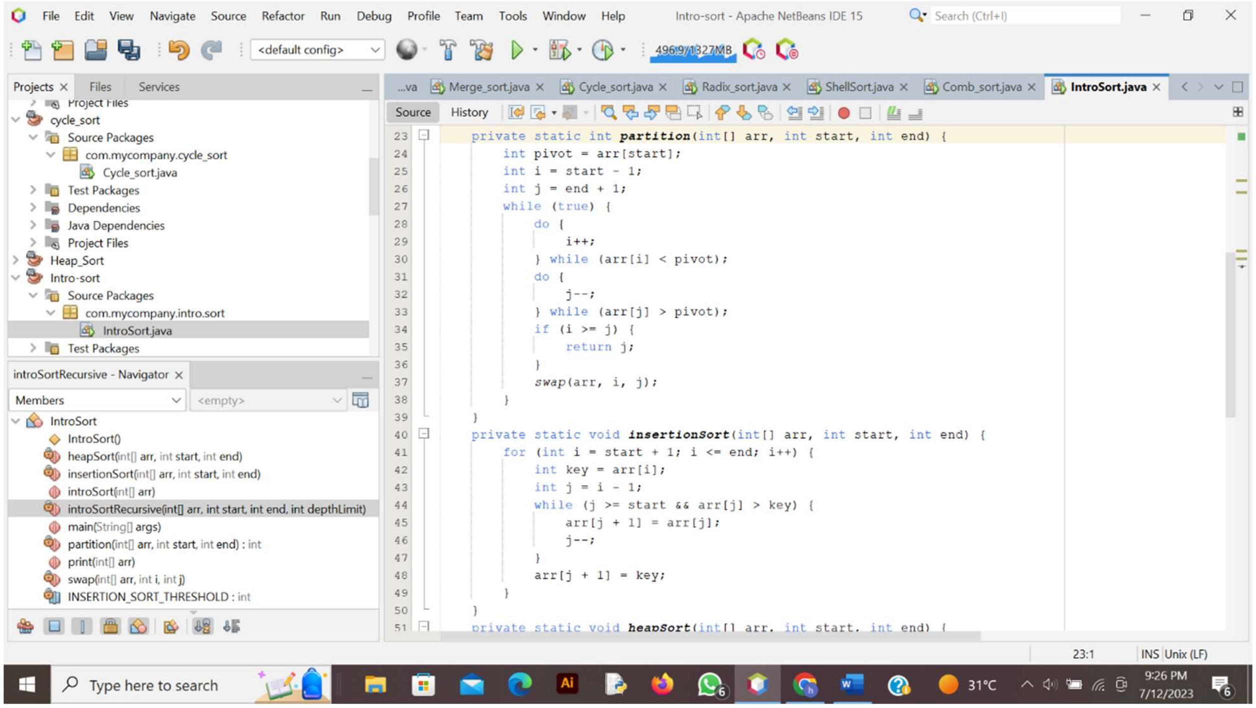
```java
        int n = arr.length;
        for (int i = 0; i < n; ++i) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
```

```
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile
Scanning for projects...

---------------------< com.mycompany:comb_sort >----------------------
Building comb_sort 1.0-SNAPSHOT
--------------------------------[ jar ]--------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ comb_sort ---
Original array:
-2 -9 -10 -5 -11 -44 -66 -1
Sorted array by Comb_sort:
-66 -44 -11 -10 -9 -5 -2 -1
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.788 s
Finished at: 2023-07-12T21:18:50-07:00
------------------------------------------------------------------------
```

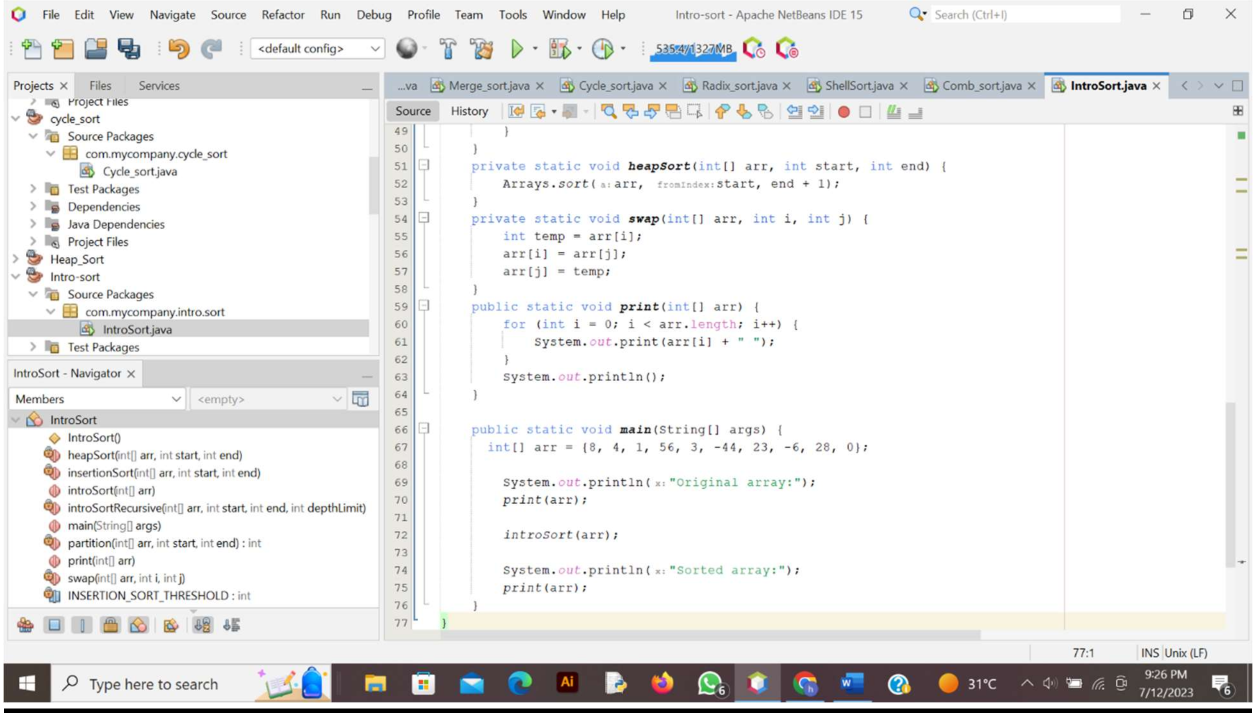
Intro Sort


Intro-sort - Apache NetBeans IDE 15
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
360.9/1327MB
Projects Files Services
Project Files
cycle_sort
Source Packages
com.mycompany.cycle_sort
Cycle_sort.java
Test Packages
Dependencies
Java Dependencies
Project Files
Heap_Sort
Intro-sort
Source Packages
com.mycompany.intro.sort
IntroSort.java
Test Packages
IntroSort.java - Navigator
Members
<empty>
IntroSort
IntroSort()
heapSort(int[] arr, int start, int end)
insertionSort(int[] arr, int start, int end)
introSort(int[] arr)
introSortRecursive(int[] arr, int start, int end, int depthLimit)
main(String[] args)
partition(int[] arr, int start, int end) : int
print(int[] arr)
swap(int[] arr, int i, int j)
INSERTION_SORT_THRESHOLD : int
...va Merge_sort.java Cycle_sort.java Radix_sort.java ShellSort.java Comb_sort.java IntroSort.java
Source History
1:1 INS Unix (LF)
Type here to search
31°C
9:26 PM
7/12/2023

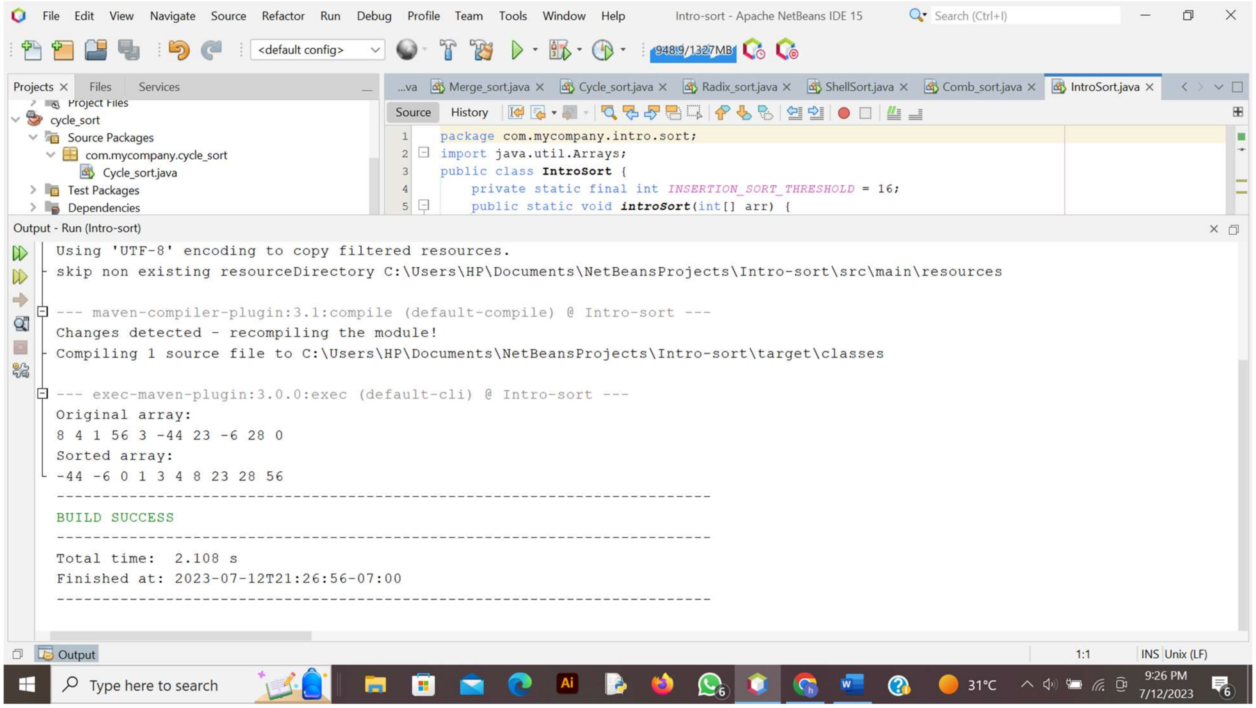
```java
package com.mycompany.intro.sort;
import java.util.Arrays;
public class IntroSort {
    private static final int INSERTION_SORT_THRESHOLD = 16;
    public static void introSort(int[] arr) {
        int depthLimit = (int) (2 * Math.log( a:arr.length));
        introSortRecursive(arr,  start:0, arr.length - 1, depthLimit);
    }
    private static void introSortRecursive(int[] arr, int start, int end, int depthLimit) {
        int size = end - start + 1;
        if (size < INSERTION_SORT_THRESHOLD) {
            insertionSort(arr, start, end);
            return;
        }
        if (depthLimit == 0) {
            heapSort(arr, start, end);
            return;
        }
        int pivotIndex = partition(arr, start, end);
        introSortRecursive(arr, start,  end:pivotIndex, depthLimit - 1);
        introSortRecursive(arr, pivotIndex + 1, end, depthLimit - 1);
    }
    private static int partition(int[] arr, int start, int end) {
        int pivot = arr[start];
        int i = start - 1;
        int j = end + 1;
        while (true) {
            do {
                i++;
```


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Intro-sort - Apache NetBeans IDE 15
Search (Ctrl+I)
<default config>
496.9/1327MB
Projects Files Services
...va Merge_sort.java Cycle_sort.java Radix_sort.java ShellSort.java Comb_sort.java IntroSort.java
Source History
Project Files
cycle_sort
Source Packages
com.mycompany.cycle_sort
Cycle_sort.java
Test Packages
Dependencies
Java Dependencies
Project Files
Heap_Sort
Intro-sort
Source Packages
com.mycompany.intro.sort
IntroSort.java
Test Packages
introSortRecursive - Navigator
Members
<empty>
IntroSort
IntroSort()
heapSort(int[] arr, int start, int end)
insertionSort(int[] arr, int start, int end)
introSort(int[] arr)
introSortRecursive(int[] arr, int start, int end, int depthLimit)
main(String[] args)
partition(int[] arr, int start, int end) : int
print(int[] arr)
swap(int[] arr, int i, int j)
INSERTION_SORT_THRESHOLD : int
23:1
INS Unix (LF)
Type here to search
31°C
9:26 PM
7/12/2023

```java
    private static int partition(int[] arr, int start, int end) {
        int pivot = arr[start];
        int i = start - 1;
        int j = end + 1;
        while (true) {
            do {
                i++;
            } while (arr[i] < pivot);
            do {
                j--;
            } while (arr[j] > pivot);
            if (i >= j) {
                return j;
            }
            swap(arr, i, j);
        }
    }
    private static void insertionSort(int[] arr, int start, int end) {
        for (int i = start + 1; i <= end; i++) {
            int key = arr[i];
            int j = i - 1;
            while (j >= start && arr[j] > key) {
                arr[j + 1] = arr[j];
                j--;
            }
            arr[j + 1] = key;
        }
    }
    private static void heapSort(int[] arr, int start, int end) {
```


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Intro-sort - Apache NetBeans IDE 15
Search (Ctrl+I)
<default config>
535.4/1327MB
Projects Files Services
...va Merge_sort.java Cycle_sort.java Radix_sort.java ShellSort.java Comb_sort.java IntroSort.java
Source History
Project Files
cycle_sort
Source Packages
com.mycompany.cycle_sort
Cycle_sort.java
Test Packages
Dependencies
Java Dependencies
Project Files
Heap_Sort
Intro-sort
Source Packages
com.mycompany.intro.sort
IntroSort.java
Test Packages
IntroSort - Navigator
Members
<empty>
IntroSort
IntroSort()
heapSort(int[] arr, int start, int end)
insertionSort(int[] arr, int start, int end)
introSort(int[] arr)
introSortRecursive(int[] arr, int start, int end, int depthLimit)
main(String[] args)
partition(int[] arr, int start, int end) : int
print(int[] arr)
swap(int[] arr, int i, int j)
INSERTION_SORT_THRESHOLD : int
77:1
INS Unix (LF)
Type here to search
31°C
9:26 PM
7/12/2023

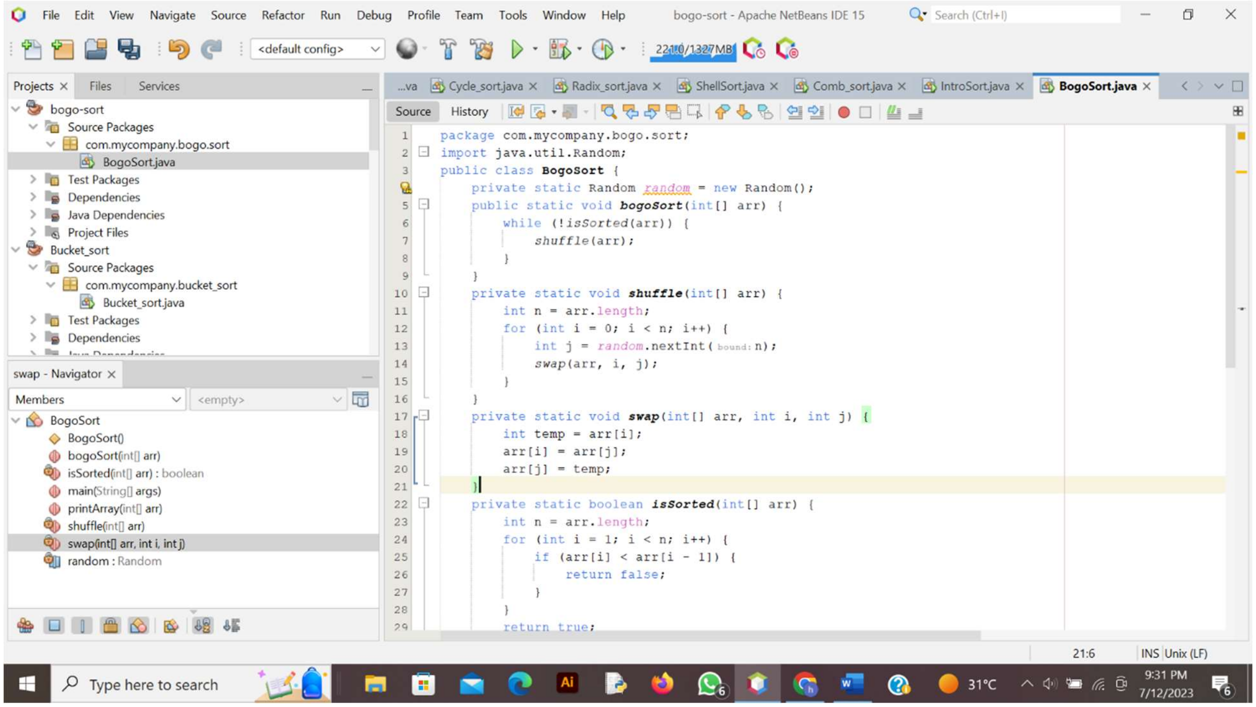
```java
        }
    }
    private static void heapSort(int[] arr, int start, int end) {
        Arrays.sort( a: arr,  fromIndex: start, end + 1);
    }
    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
    public static void print(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
      int[] arr = {8, 4, 1, 56, 3, -44, 23, -6, 28, 0};

        System.out.println( x: "Original array:");
        print(arr);

        introSort(arr);

        System.out.println( x: "Sorted array:");
        print(arr);
    }
}
```



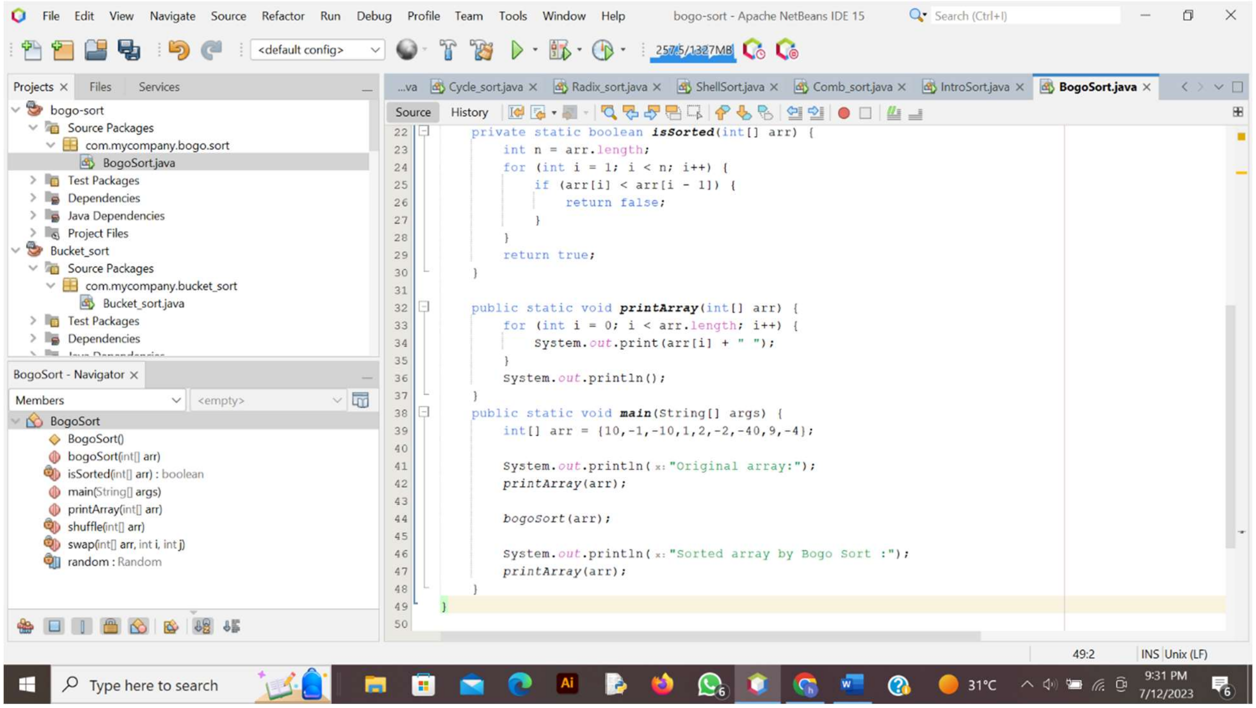
```
package com.mycompany.intro.sort;
import java.util.Arrays;
public class IntroSort {
    private static final int INSERTION_SORT_THRESHOLD = 16;
    public static void introSort(int[] arr) {
```

```
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\HP\Documents\NetBeansProjects\Intro-sort\src\main\resources

--- maven-compiler-plugin:3.1:compile (default-compile) @ Intro-sort ---
Changes detected - recompiling the module!
Compiling 1 source file to C:\Users\HP\Documents\NetBeansProjects\Intro-sort\target\classes

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Intro-sort ---
Original array:
8 4 1 56 3 -44 23 -6 28 0
Sorted array:
-44 -6 0 1 3 4 8 23 28 56
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.108 s
Finished at: 2023-07-12T21:26:56-07:00
------------------------------------------------------------------------
```

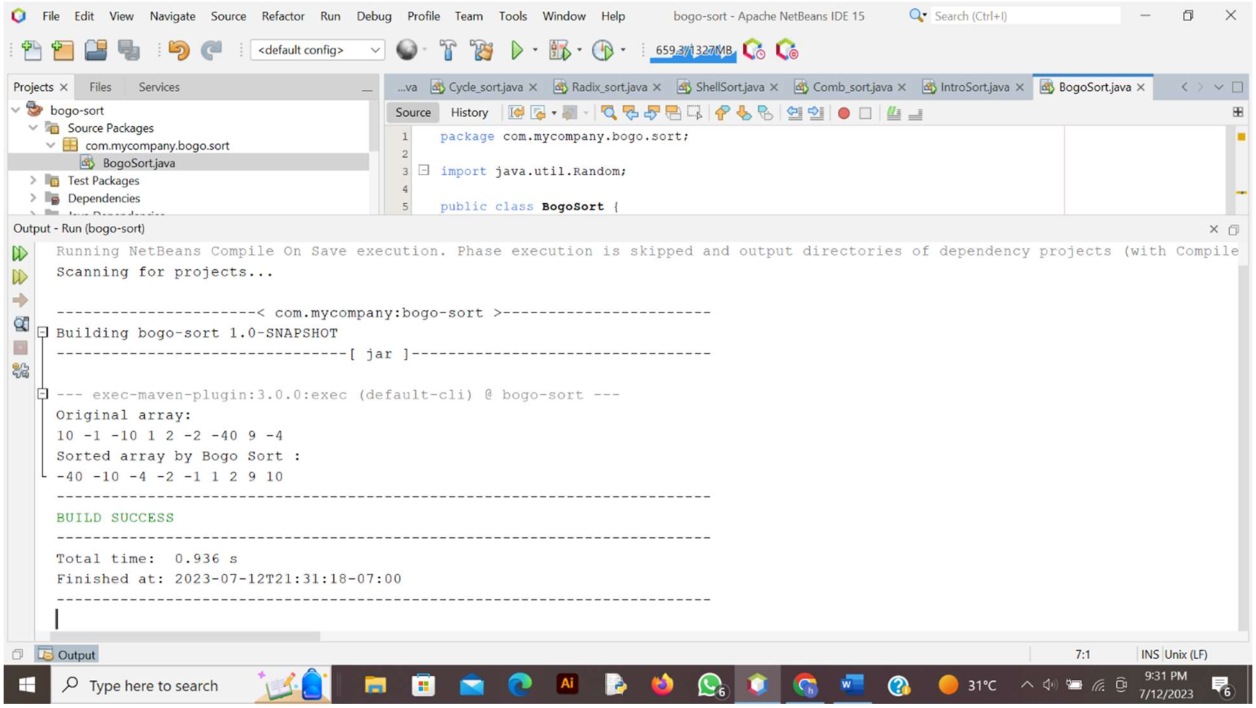
Bogo Sort



```
package com.mycompany.bogo.sort;
import java.util.Random;
public class BogoSort {
    private static Random random = new Random();
    public static void bogoSort(int[] arr) {
        while (!isSorted(arr)) {
            shuffle(arr);
        }
    }
    private static void shuffle(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n; i++) {
            int j = random.nextInt( bound: n);
            swap(arr, i, j);
        }
    }
    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
    private static boolean isSorted(int[] arr) {
        int n = arr.length;
        for (int i = 1; i < n; i++) {
            if (arr[i] < arr[i - 1]) {
                return false;
            }
        }
        return true;
```



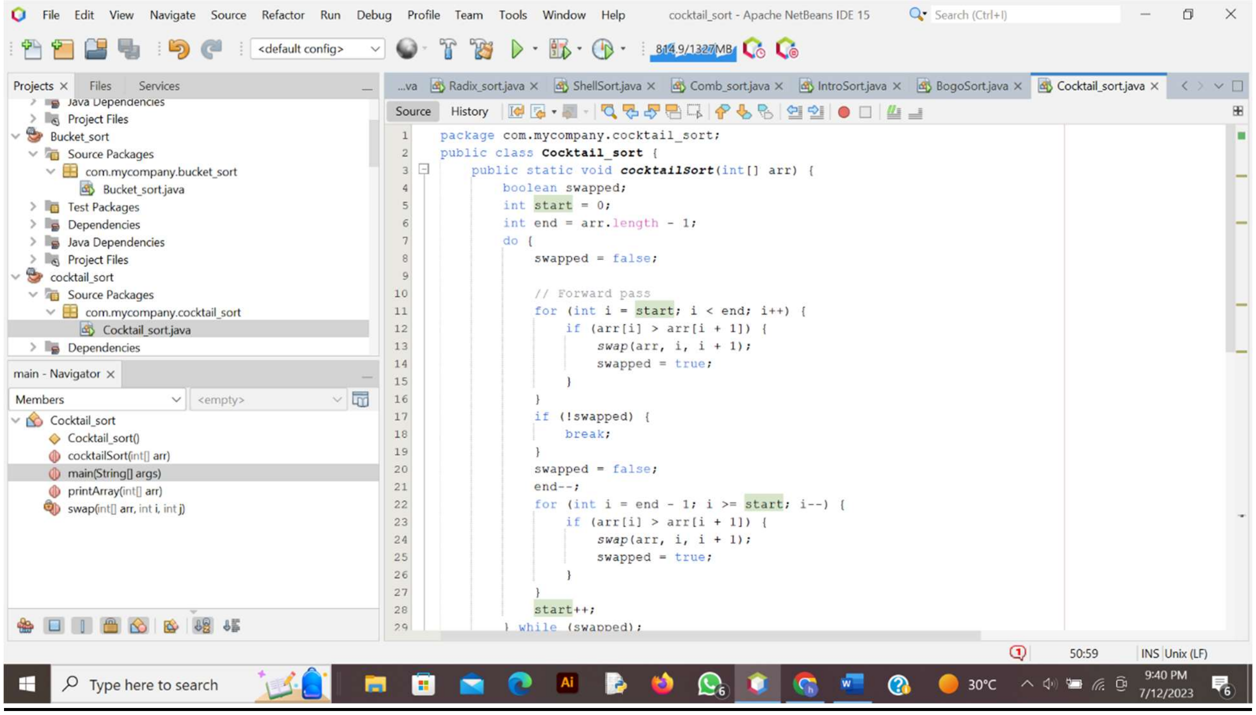
```java
    private static boolean isSorted(int[] arr) {
        int n = arr.length;
        for (int i = 1; i < n; i++) {
            if (arr[i] < arr[i - 1]) {
                return false;
            }
        }
        return true;
    }

    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] arr = {10,-1,-10,1,2,-2,-40,9,-4};

        System.out.println( x: "Original array:");
        printArray(arr);

        bogoSort(arr);

        System.out.println( x: "Sorted array by Bogo Sort :");
        printArray(arr);
    }
}
```



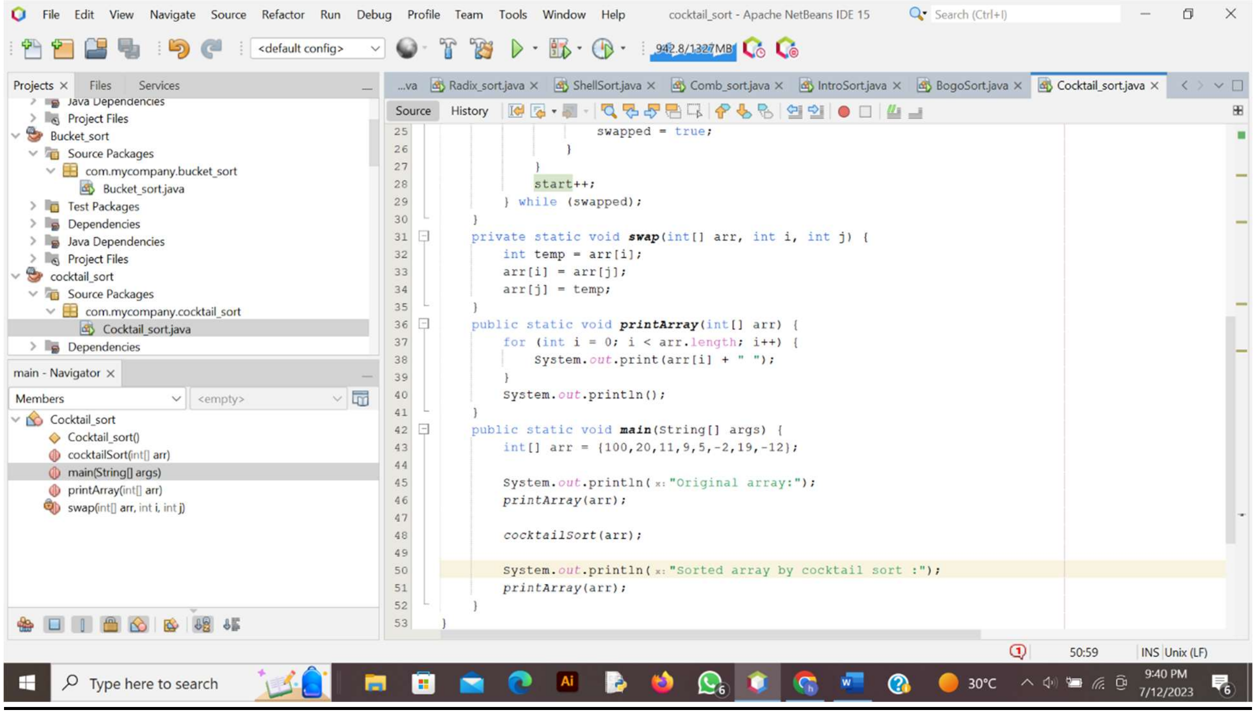
```java
package com.mycompany.bogo.sort;

import java.util.Random;

public class BogoSort {
```

```
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile
Scanning for projects...

---------------------< com.mycompany:bogo-sort >----------------------
Building bogo-sort 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ bogo-sort ---
Original array:
10 -1 -10 1 2 -2 -40 9 -4
Sorted array by Bogo Sort :
-40 -10 -4 -2 -1 1 2 9 10
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.936 s
Finished at: 2023-07-12T21:31:18-07:00
------------------------------------------------------------------------
```

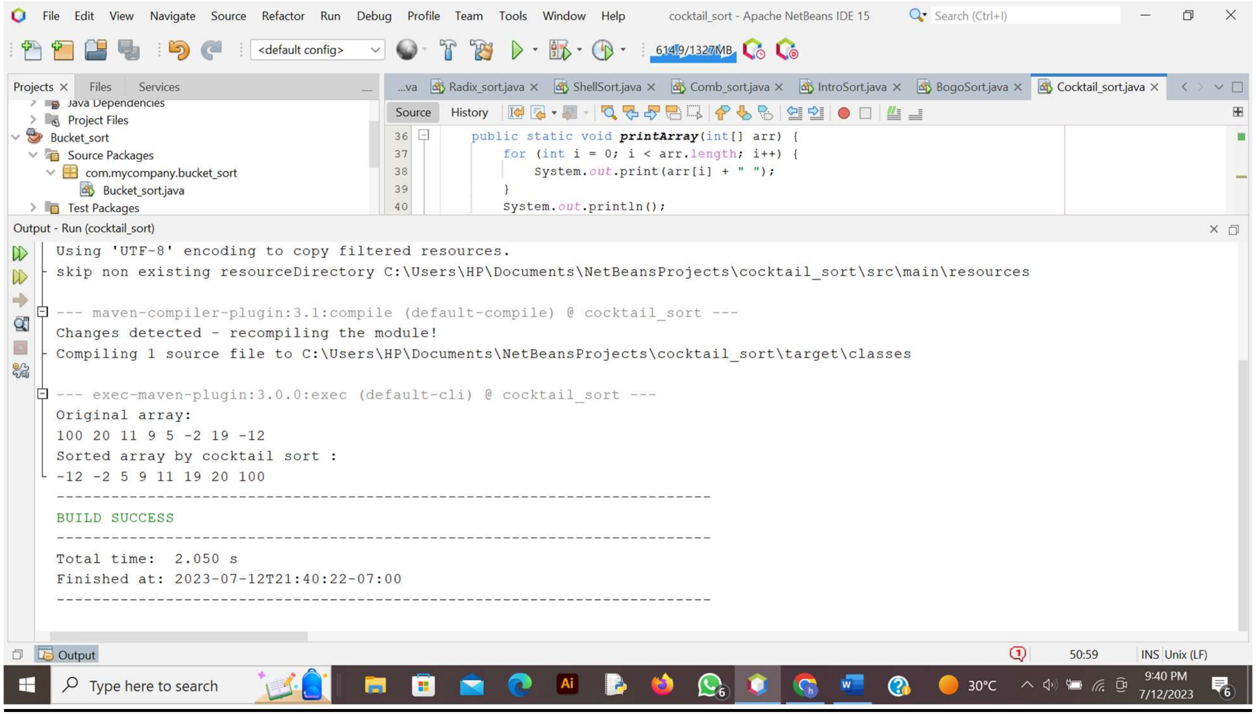
Cocktail Sort



```java
package com.mycompany.cocktail_sort;
public class Cocktail_sort {
    public static void cocktailSort(int[] arr) {
        boolean swapped;
        int start = 0;
        int end = arr.length - 1;
        do {
            swapped = false;

            // Forward pass
            for (int i = start; i < end; i++) {
                if (arr[i] > arr[i + 1]) {
                    swap(arr, i, i + 1);
                    swapped = true;
                }
            }
            if (!swapped) {
                break;
            }
            swapped = false;
            end--;
            for (int i = end - 1; i >= start; i--) {
                if (arr[i] > arr[i + 1]) {
                    swap(arr, i, i + 1);
                    swapped = true;
                }
            }
            start++;
        } while (swapped);
```



```java
                    swapped = true;
                }
            }
            start++;
        } while (swapped);
    }
    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] arr = {100,20,11,9,5,-2,19,-12};

        System.out.println( x: "Original array:");
        printArray(arr);

        cocktailSort(arr);

        System.out.println( x: "Sorted array by cocktail sort :");
        printArray(arr);
    }
}
```

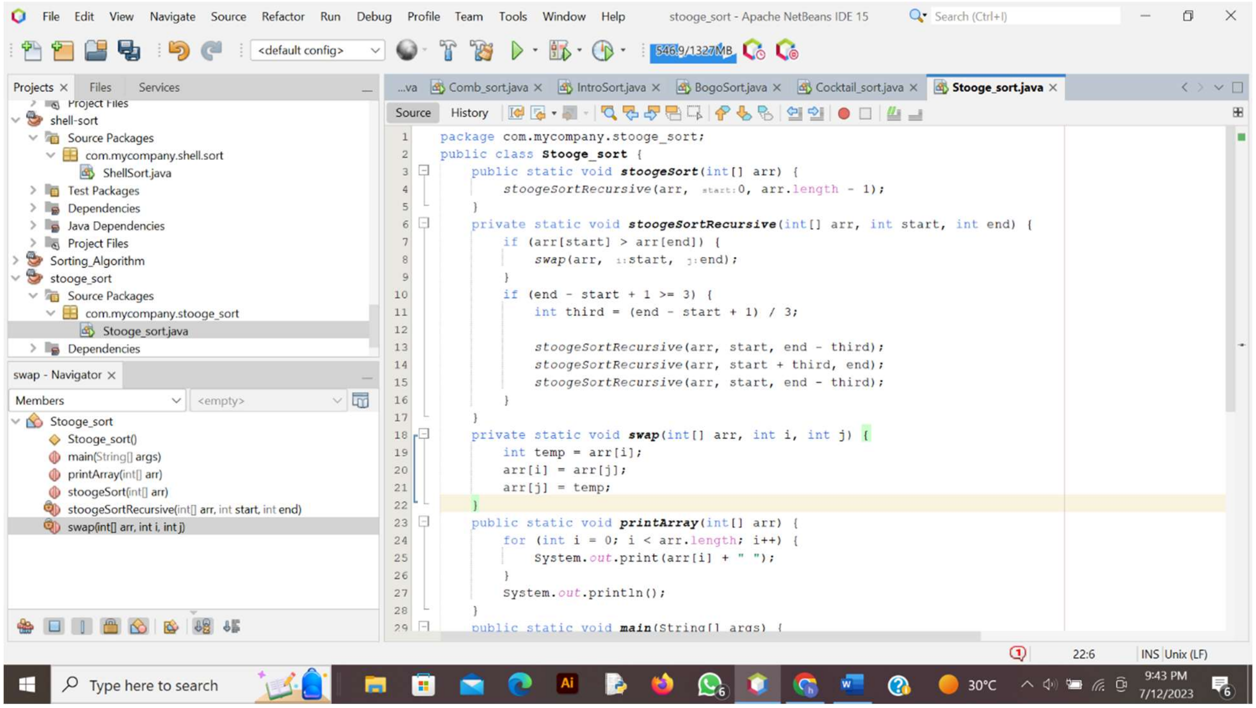


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
cocktail_sort - Apache NetBeans IDE 15
Search (Ctrl+I)
<default config>
614.9/1327MB
Projects Files Services
Java Dependencies
Project Files
Bucket_sort
Source Packages
com.mycompany.bucket_sort
Bucket_sort.java
Test Packages
...va Radix_sort.java ShellSort.java Comb_sort.java IntroSort.java BogoSort.java Cocktail_sort.java
Source History
Output - Run (cocktail_sort)
Output
50:59
INS Unix (LF)
Type here to search
30°C
9:40 PM
7/12/2023

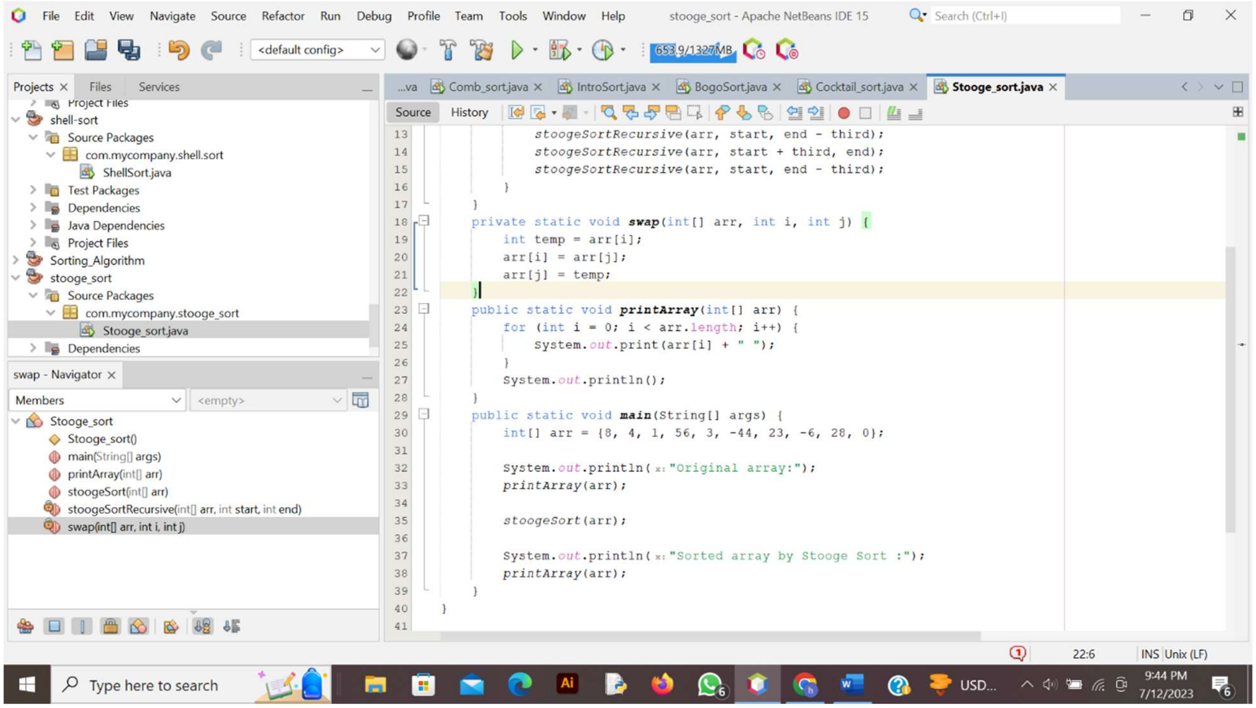
```java
    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
```

```
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\HP\Documents\NetBeansProjects\cocktail_sort\src\main\resources

--- maven-compiler-plugin:3.1:compile (default-compile) @ cocktail_sort ---
Changes detected - recompiling the module!
Compiling 1 source file to C:\Users\HP\Documents\NetBeansProjects\cocktail_sort\target\classes

--- exec-maven-plugin:3.0.0:exec (default-cli) @ cocktail_sort ---
Original array:
100 20 11 9 5 -2 19 -12
Sorted array by cocktail sort :
-12 -2 5 9 11 19 20 100
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.050 s
Finished at: 2023-07-12T21:40:22-07:00
------------------------------------------------------------------------
```

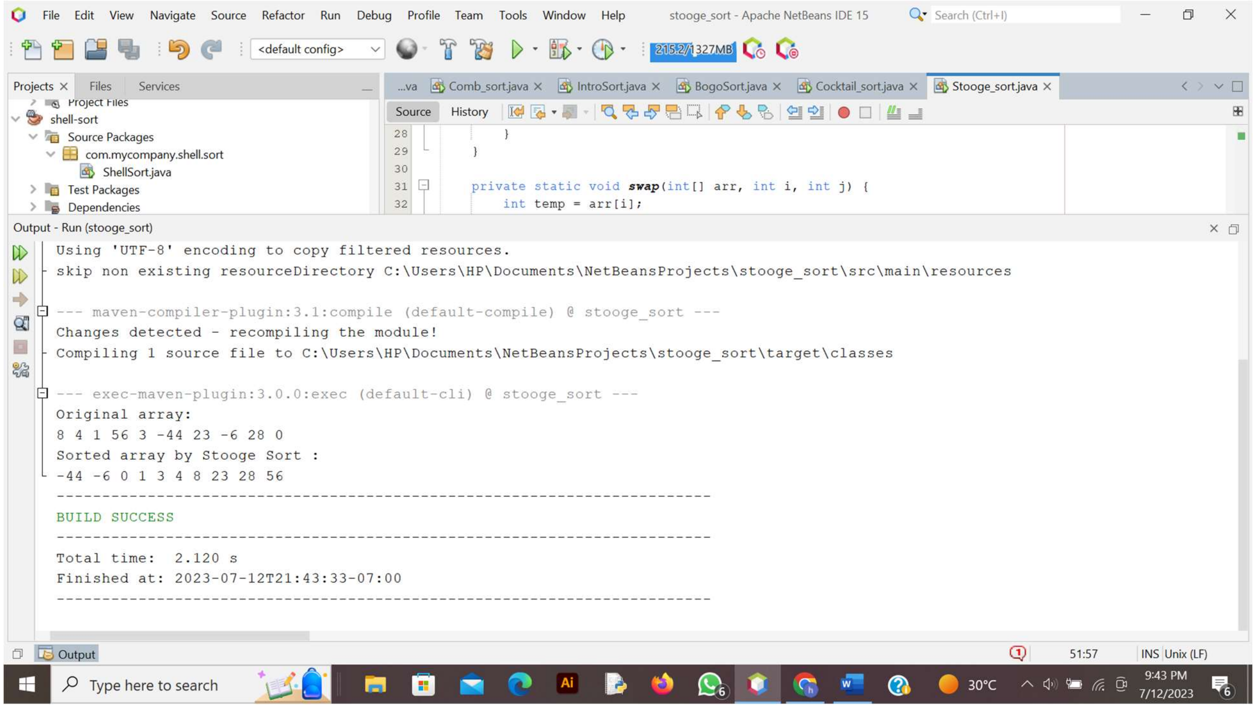
Stooge Sort



```java
package com.mycompany.stooge_sort;
public class Stooge_sort {
    public static void stoogeSort(int[] arr) {
        stoogeSortRecursive(arr, start:0, arr.length - 1);
    }
    private static void stoogeSortRecursive(int[] arr, int start, int end) {
        if (arr[start] > arr[end]) {
            swap(arr, i:start, j:end);
        }
        if (end - start + 1 >= 3) {
            int third = (end - start + 1) / 3;

            stoogeSortRecursive(arr, start, end - third);
            stoogeSortRecursive(arr, start + third, end);
            stoogeSortRecursive(arr, start, end - third);
        }
    }
    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
```



```java
                stoogeSortRecursive(arr, start, end - third);
                stoogeSortRecursive(arr, start + third, end);
                stoogeSortRecursive(arr, start, end - third);
            }
        }
        private static void swap(int[] arr, int i, int j) {
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
        public static void printArray(int[] arr) {
            for (int i = 0; i < arr.length; i++) {
                System.out.print(arr[i] + " ");
            }
            System.out.println();
        }
        public static void main(String[] args) {
            int[] arr = {8, 4, 1, 56, 3, -44, 23, -6, 28, 0};

            System.out.println( x: "Original array:");
            printArray(arr);

            stoogeSort(arr);

            System.out.println( x: "Sorted array by Stooge Sort :");
            printArray(arr);
        }
    }
```

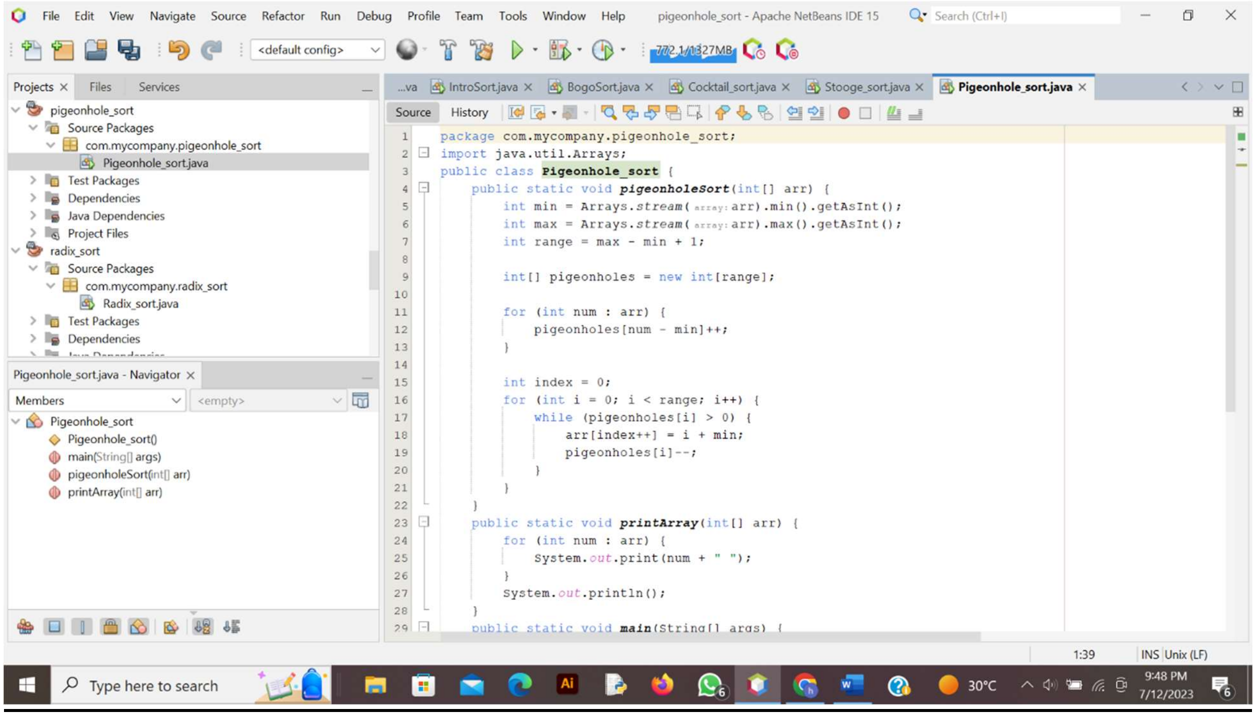


```java
            }
        }

        private static void swap(int[] arr, int i, int j) {
            int temp = arr[i];
```

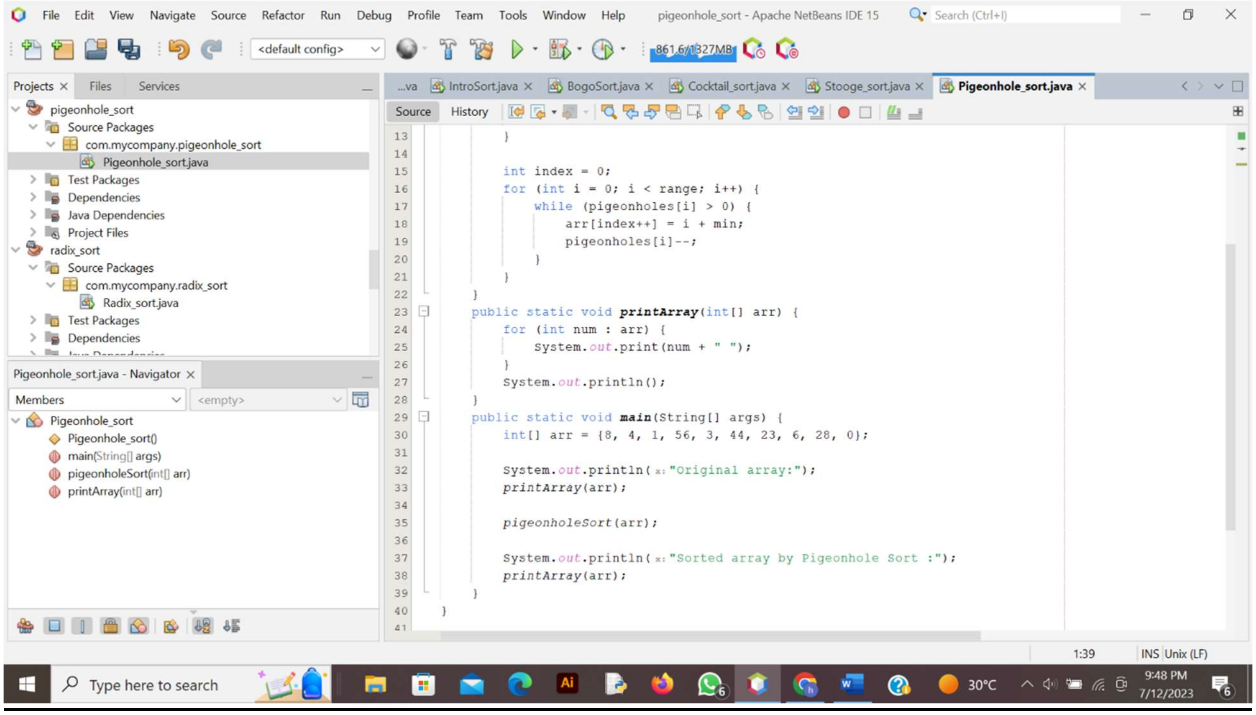
Output - Run (stooge_sort)

```
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\HP\Documents\NetBeansProjects\stooge_sort\src\main\resources

--- maven-compiler-plugin:3.1:compile (default-compile) @ stooge_sort ---
Changes detected - recompiling the module!
Compiling 1 source file to C:\Users\HP\Documents\NetBeansProjects\stooge_sort\target\classes

--- exec-maven-plugin:3.0.0:exec (default-cli) @ stooge_sort ---
Original array:
8 4 1 56 3 -44 23 -6 28 0
Sorted array by Stooge Sort :
-44 -6 0 1 3 4 8 23 28 56
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.120 s
Finished at: 2023-07-12T21:43:33-07:00
------------------------------------------------------------------------
```

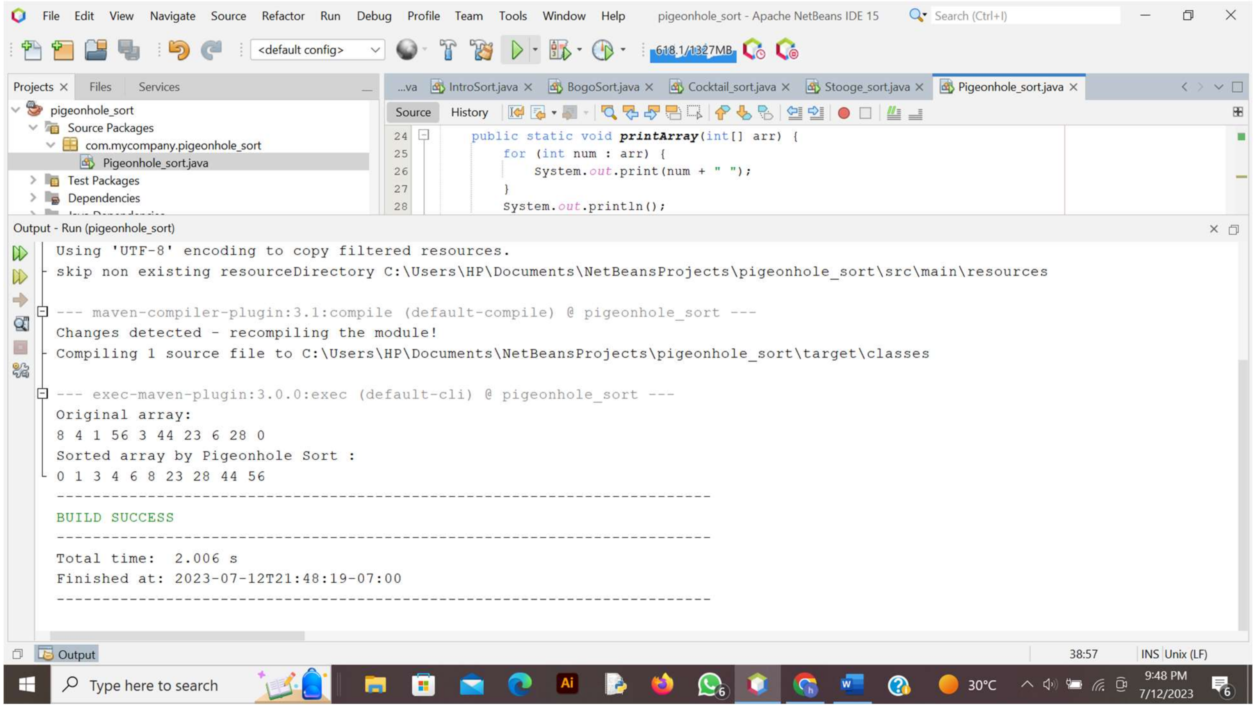
Pigeonhole Sort



```java
package com.mycompany.pigeonhole_sort;
import java.util.Arrays;
public class Pigeonhole_sort {
    public static void pigeonholeSort(int[] arr) {
        int min = Arrays.stream( array: arr).min().getAsInt();
        int max = Arrays.stream( array: arr).max().getAsInt();
        int range = max - min + 1;

        int[] pigeonholes = new int[range];

        for (int num : arr) {
            pigeonholes[num - min]++;
        }

        int index = 0;
        for (int i = 0; i < range; i++) {
            while (pigeonholes[i] > 0) {
                arr[index++] = i + min;
                pigeonholes[i]--;
            }
        }
    }
    public static void printArray(int[] arr) {
        for (int num : arr) {
            System.out.print(num + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
```



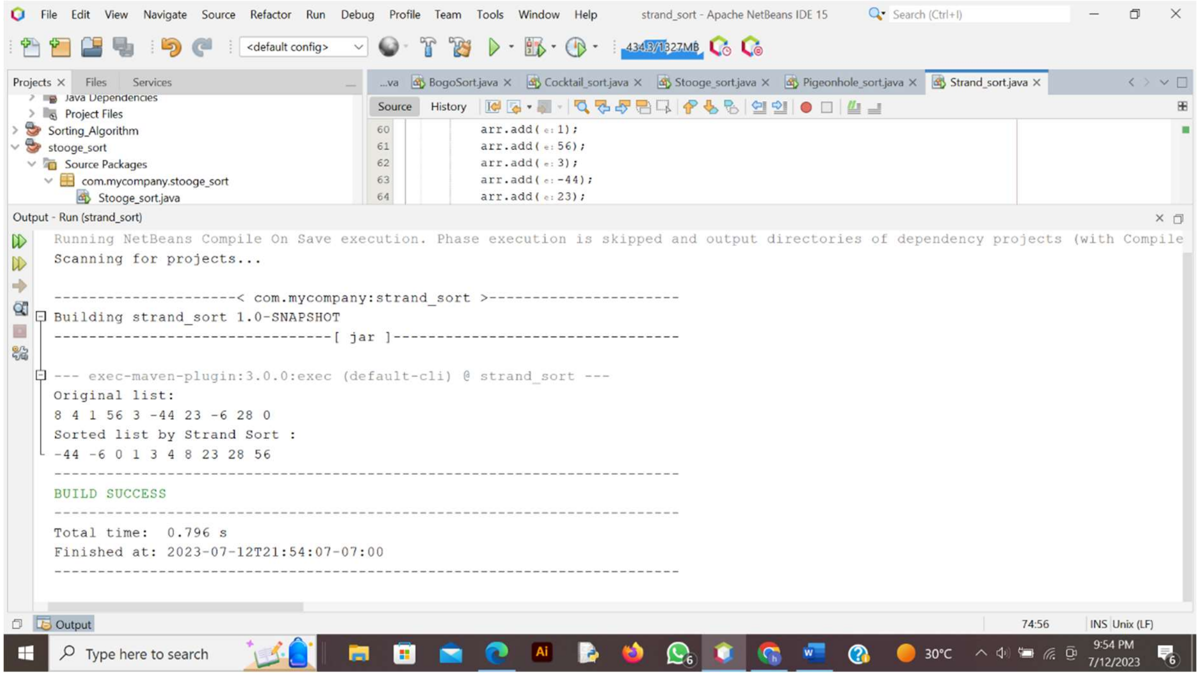
```java
        }

        int index = 0;
        for (int i = 0; i < range; i++) {
            while (pigeonholes[i] > 0) {
                arr[index++] = i + min;
                pigeonholes[i]--;
            }
        }
    }
    public static void printArray(int[] arr) {
        for (int num : arr) {
            System.out.print(num + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] arr = {8, 4, 1, 56, 3, 44, 23, 6, 28, 0};

        System.out.println( x: "Original array:");
        printArray(arr);

        pigeonholeSort(arr);

        System.out.println( x: "Sorted array by Pigeonhole Sort :");
        printArray(arr);
    }
}
```


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
pigeonhole_sort - Apache NetBeans IDE 15
Search (Ctrl+I)
<default config>
618.1/1327MB
Projects Files Services
pigeonhole_sort
Source Packages
com.mycompany.pigeonhole_sort
Pigeonhole_sort.java
Test Packages
Dependencies
...va IntroSort.java BogoSort.java Cocktail_sort.java Stooge_sort.java Pigeonhole_sort.java
Source History
Output - Run (pigeonhole_sort)
Output
38:57
INS Unix (LF)
Type here to search
30°C
9:48 PM
7/12/2023

```
    public static void printArray(int[] arr) {
        for (int num : arr) {
            System.out.print(num + " ");
        }
        System.out.println();
```

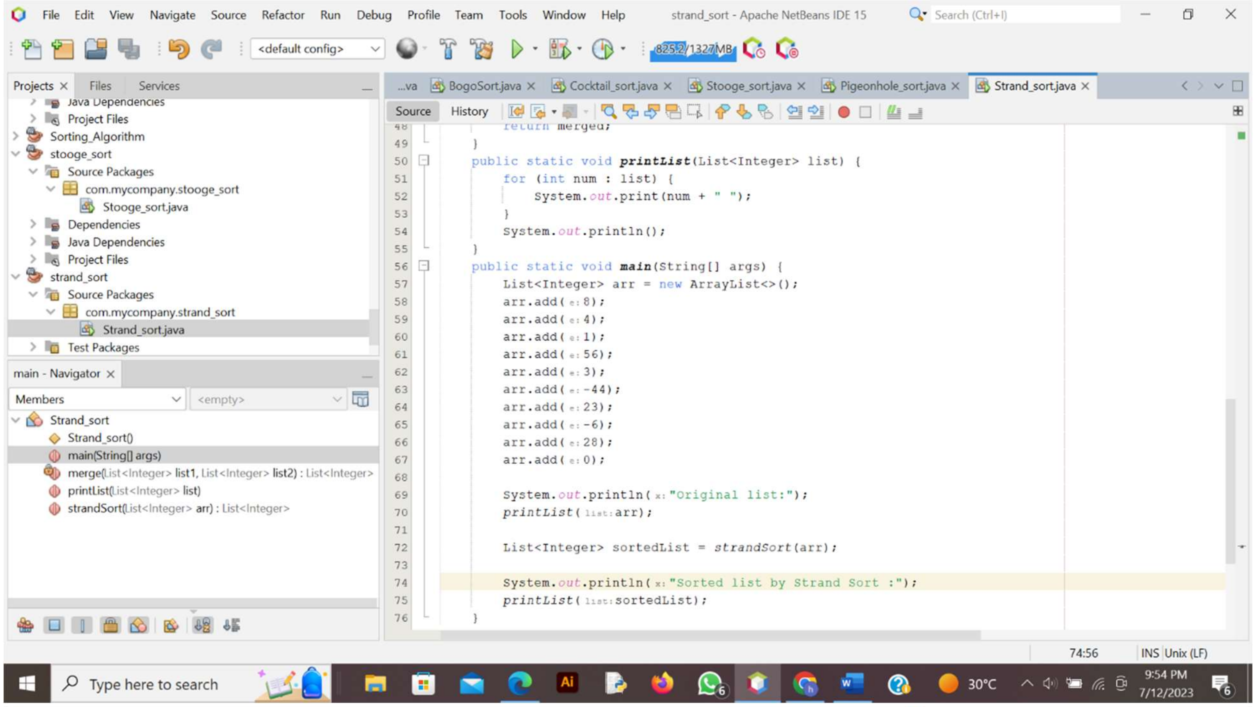
```
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\HP\Documents\NetBeansProjects\pigeonhole_sort\src\main\resources

--- maven-compiler-plugin:3.1:compile (default-compile) @ pigeonhole_sort ---
Changes detected - recompiling the module!
Compiling 1 source file to C:\Users\HP\Documents\NetBeansProjects\pigeonhole_sort\target\classes

--- exec-maven-plugin:3.0.0:exec (default-cli) @ pigeonhole_sort ---
Original array:
8 4 1 56 3 44 23 6 28 0
Sorted array by Pigeonhole Sort :
0 1 3 4 6 8 23 28 44 56
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.006 s
Finished at: 2023-07-12T21:48:19-07:00
------------------------------------------------------------------------
```

Strand Sort

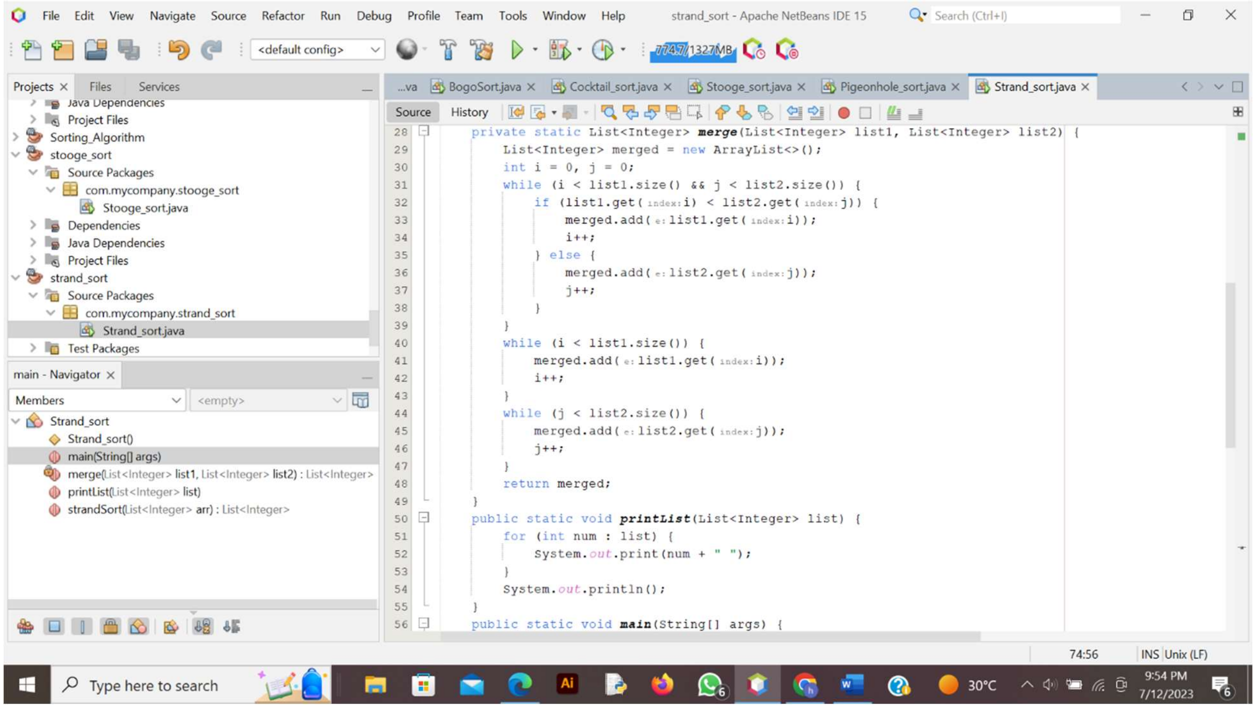

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
strand_sort - Apache NetBeans IDE 15
Search (Ctrl+I)
<default config>
434.3/1327MB
Projects Files Services
Java Dependencies
Project Files
Sorting_Algorithm
stooge_sort
Source Packages
com.mycompany.stooge_sort
Stooge_sort.java
...va BogoSort.java Cocktail_sort.java Stooge_sort.java Pigeonhole_sort.java Strand_sort.java
Source History
Output - Run (strand_sort)
Output
74:56
INS Unix (LF)
Type here to search
30°C
9:54 PM
7/12/2023

```
        arr.add( e: 1);
        arr.add( e: 56);
        arr.add( e: 3);
        arr.add( e: -44);
        arr.add( e: 23);
```

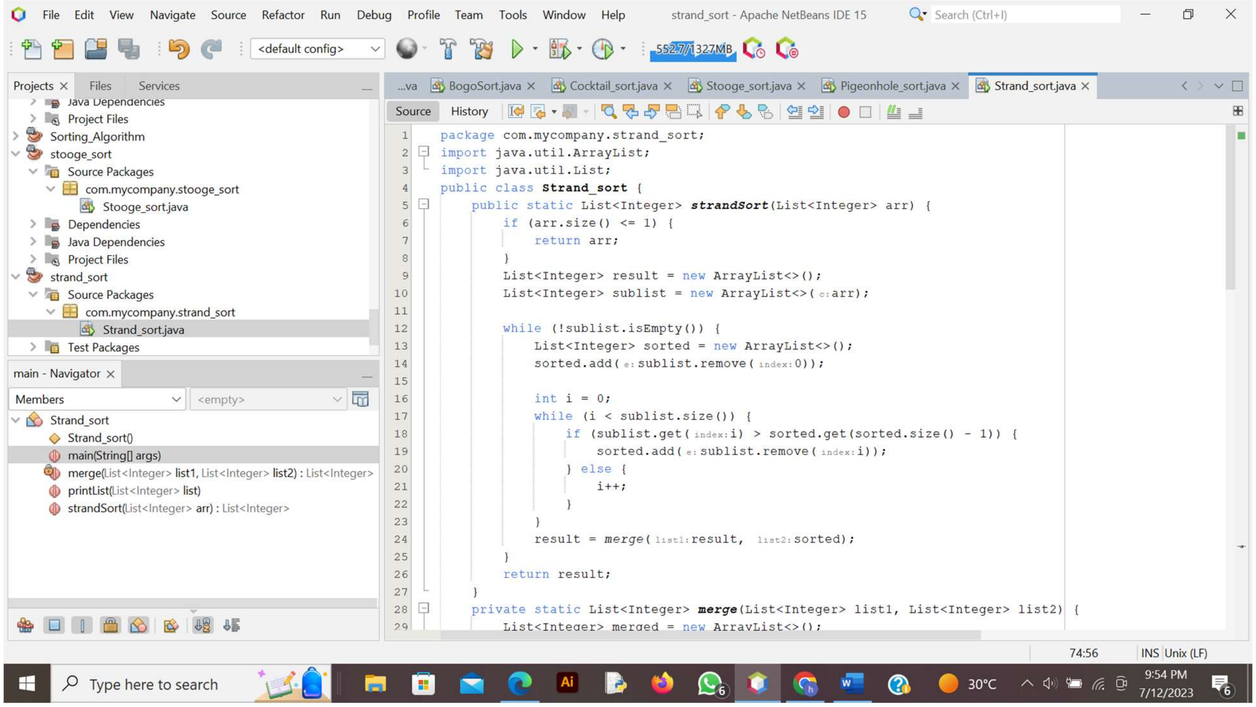
```
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile
Scanning for projects...

--------------------< com.mycompany:strand_sort >---------------------
Building strand_sort 1.0-SNAPSHOT
--------------------------------[ jar ]--------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ strand_sort ---
Original list:
8 4 1 56 3 -44 23 -6 28 0
Sorted list by Strand Sort :
-44 -6 0 1 3 4 8 23 28 56
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.796 s
Finished at: 2023-07-12T21:54:07-07:00
------------------------------------------------------------------------
```



```java
        return merged;
    }
    public static void printList(List<Integer> list) {
        for (int num : list) {
            System.out.print(num + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        List<Integer> arr = new ArrayList<>();
        arr.add(8);
        arr.add(4);
        arr.add(1);
        arr.add(56);
        arr.add(3);
        arr.add(-44);
        arr.add(23);
        arr.add(-6);
        arr.add(28);
        arr.add(0);

        System.out.println("Original list:");
        printList(arr);

        List<Integer> sortedList = strandSort(arr);

        System.out.println("Sorted list by Strand Sort :");
        printList(sortedList);
    }
```



```java
    private static List<Integer> merge(List<Integer> list1, List<Integer> list2) {
        List<Integer> merged = new ArrayList<>();
        int i = 0, j = 0;
        while (i < list1.size() && j < list2.size()) {
            if (list1.get(i) < list2.get(j)) {
                merged.add(list1.get(i));
                i++;
            } else {
                merged.add(list2.get(j));
                j++;
            }
        }
        while (i < list1.size()) {
            merged.add(list1.get(i));
            i++;
        }
        while (j < list2.size()) {
            merged.add(list2.get(j));
            j++;
        }
        return merged;
    }
    public static void printList(List<Integer> list) {
        for (int num : list) {
            System.out.print(num + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
```



```java
package com.mycompany.strand_sort;
import java.util.ArrayList;
import java.util.List;
public class Strand_sort {
    public static List<Integer> strandSort(List<Integer> arr) {
        if (arr.size() <= 1) {
            return arr;
        }
        List<Integer> result = new ArrayList<>();
        List<Integer> sublist = new ArrayList<>( c:arr);

        while (!sublist.isEmpty()) {
            List<Integer> sorted = new ArrayList<>();
            sorted.add( e:sublist.remove( index:0));

            int i = 0;
            while (i < sublist.size()) {
                if (sublist.get( index:i) > sorted.get(sorted.size() - 1)) {
                    sorted.add( e:sublist.remove( index:i));
                } else {
                    i++;
                }
            }
            result = merge( list1:result,  list2:sorted);
        }
        return result;
    }
    private static List<Integer> merge(List<Integer> list1, List<Integer> list2) {
        List<Integer> merged = new ArrayList<>();
```