FILE NAME: BSCS2_Marasigan_Sorting.java && BSCS2_Marasigan_Sorting.pdf

SOURCE CODE

```java
package sorting;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
public class BSCS2_Marasigan_Sorting
{
    static Scanner in = new Scanner(System.in);
    public static void main(String[] args)
    {
        Tools.Head();
        System.out.print("   Enter the size of an array:\t");
        int size = in.nextInt();
        int[] A= new int[size], B = new int[size];
                        //backup for sorting repetition
        System.out.print("   Enter " + size + " values: ");
        for (int count = 0; count<size; count++)
        {
            A[count] = in.nextInt();
            B[count] = A[count];
        }
        Tools.Space();
        OptionLoop(A, B);
        in.close();
    }

    static void OptionLoop(int[] A, int[] B)
    {
        Tools.Menu();
        int choice = in.nextInt();
        Tools.Space();
        switch (choice)
        {
        case 1: new Bubble(A); break;
        case 2: new Insertion(A); break;
        case 3: new Selection(A); break;
        case 4: Tools.End(); break;
        default: OptionLoop(A, B);
        }
        Tools.PrintInFormat1(2, A, "+--SORTED ARRAY--");
        Tools.PrintInFormat3(A);
        A = unsortedRetriever(A, B);
        Tools.Space();
        OptionLoop(A, B);
    }
    static int[] unsortedRetriever(int []sorted, int[]unsorted )
    {
        for (int i = 0; i<sorted.length; i++)
        {
            sorted[i] = unsorted[i];
        }
        return sorted;
```

```java
        }
}

/*I apply using classes too as a matter of practicing what I learned
in Intermediate Programming ^_^ */
//I separated the prints for neatness of the code
class Bubble
{
    Bubble(int[] A)
    {
        int swap, j, n;
        int pass = 1;
        n = A.length;

        Tools.PrintInFormat1(2, A, "+--[ BUBBLE ]----");
        while (Tools.sortingChecker(A, 0, A.length-1))
        {// originally, condition (i < n) is replaced for efficiency
            Tools.PassFormat(pass, A);
            for (j=0; j < n-1; j++)
            {
                if (A[j] > A[j+1])
                {
                    swap = A[j];
                    A[j] = A[j+1];
                    A[j+1] = swap;
                }
                CustomFormat(A);
            }
            pass++;
        }
        Tools.PrintInFormat3(A);
    }

    static void CustomFormat(int[] array)
    {
        System.out.print("\t|");
        Tools.PrintArray(array);
        System.out.print("|");
        Tools.Space();
    }
}

class Insertion
{
    Insertion(int[] A)
    {
        int temp = 0;
        Tools.PrintInFormat1(2, A, "+--[ INSERTION ]-");
        for (int range = 1;Tools.sortingChecker(A, 0, A.length-1); range++)
        {//(range++) picks the next element for checking
            if (Tools.sortingChecker(A, 0, range)) //checks if sorted
            {
                temp = A[range];
                for (int a = 0; a<=range; a++)
```

```java
                {//comparison to all elements
                    if (temp<A[a])
                    {
                        A = Tools.shiftElements(A, a, range); //shifts the elements
                        A[a] = temp; //inserts the value
                        break;
                    }
                }
            }
            Tools.PassFormat(range, A);
            Tools.PrintInFormat2(A, range);
        }
        Tools.PrintInFormat3(A);
    }
}

class Selection
{
    Selection(int[] A)
    {
        int swap;
        int n = A.length;
        Tools.PrintInFormat1(2, A, "+--[ SELECTION ]-");
        for (int i =0; Tools.sortingChecker(A, 0, n-1); i++)
        {
            int min = Tools.findMin(A, i, n-1);
            int index = Tools.indexFinder(min, A, i, n-1);
            swap = A[i];
            A[i] = min;
            A[index] = swap;

            Tools.PassFormat(i+1, A);
            Tools.PrintInFormat2(A, i+1);
        }
        Tools.PrintInFormat3(A);
    }
}

class Tools
{
    //Sorting Tools
    static boolean sortingChecker(int[] array, int start, int end)
    {
        for (; start < end; start++)
        {
            if (array[start+1] < array[start] )
            {
                return true;//means it's not sorted yet
            }
        }
        return false; //break
        /* Note: since this checker that is responsible for breaks is working properly
        I decided to replace the algorithm a little to make use of its effectiveness*/
    }
```

```java
static void passIndicator(int pass)
{
    String s = Integer.toString(pass);
    char last = s.charAt(s.length()-1); //uses the first digit as reference
            //example: 21 ; 1 will be reference to make it "21st" instead of "21th"
    if (last == '1' && pass != 11 )
    {
        System.out.print(pass + "st\tPASS");
    }
    else if (last == '2'&& pass != 12)
    {
        System.out.print(pass + "nd\tPASS");
    }
    else if (last == '3'&& pass != 13)
    {
        System.out.print(pass + "rd\tPASS");
    }
    else
    {
        System.out.print(pass + "th\tPASS");
    }
}
static int indexFinder(int value, int[] Array, int start, int end)
{
    int b;
    for(b = start; b<end; b++)
    {
        if(value == Array[b])
        {
            break;
        }
    }
    return b;
}
static int[] shiftElements(int[]Array, int start, int end)
{
    for (; end>start; end--)
    {
        Array[end]=Array[end-1];
    }
    return Array;
}
static int findMin(int[] A, int start, int end)
{//          controls the number of elements to be checked[start, end]
    int min=0; int isLess=0;
    for (int a = start; a<=end; a++)
    {
        for (int b = start; b<=end; b++)
        {
            if (A[a] < A[b])
            {
                isLess++;
            }
        }
    }
```

```java
                if (isLess == (end-start))
                {
                    min = A[a];
                    break;
                }
                else
                {
                    isLess = 0;
                }
            }
        return min;
    }
    //Designing Tools
    static void PrintInFormat1(int start, int[] array, String name)
    {
        System.out.print("\t" +name);
        Tools.PrintLine(array.length, start);
        Tools.Space();
        System.out.print("\t|");
        Tools.PrintArray(array);
        System.out.print("|");
        Tools.Space();
//      System.out.print("\t+");
//      Tools.PrintLine(array.length, 0);
//      Tools.Space();
    }
    static void PrintInFormat2(int[] array, int pass)
    {
        System.out.print("\t|");
        Tools.PrintArray(array);
        System.out.print("| ");
        Tools.Space();
    }
    static void PassFormat(int pass, int[] array)
    {
        System.out.print("\t+--");
        Tools.passIndicator(pass);
        System.out.print("-----");
        Tools.PrintLine(array.length, 2);
        Tools.Space();
    }
    static void PrintInFormat3(int[] array)
    {
        System.out.print("\t+");
        Tools.PrintLine(array.length, 0);
        Tools.Space();
    }
    static void PrintArray(int[] array)
    {
        try
        {
            System.out.print("\t");
            for (int i = 0; i < array.length; i++)
            {
```

```java
                    System.out.print(array[i] + "\t");
                    TimeUnit.MILLISECONDS.sleep(100);
            }
        }
        catch (Exception e)
        {

        }
}
static void Space()
{
    System.out.println();
}
static void PrintLine(int limit, int start)
{
    try{
        for (int i = start-1; i < limit; i++)
        {
            if (i == limit-1)
            {
                for (int count = 0; count<7;count++)
                {
                System.out.print("-");
                TimeUnit.MILLISECONDS.sleep(25);
                }
                System.out.print("+");
            }
            else
            {
                for (int count = 0; count<8;count++)
                {
                System.out.print("-");
                TimeUnit.MILLISECONDS.sleep(25);
                }
            }
        }
    }
    catch(Exception e){

    }
}
static void Head()
{//https://patorjk.com/software/taag/
    String sorting = "      ___        ___       ___      ___                 ___         ___     ___     \r\n"
        + "        /\\  \\      /\\  \\      /\\  \\    /\\  \\      ___      /\\__\\      /\\  \\   \\r\n"
        + "       /::\\  \\     /::\\  \\     /::\\  \\   \\:\\  \\     /\\  \\    /::|  |     /::\\  \\   \\r\n"
        + "      /:/\\ \\  \\   /:/\\:\\  \\   /:/\\:\\  \\   \\:\\  \\    \\:\\  \\  /:|:|  |    /:/\\:\\  \\  \\r\n"
        + "     _\\:\\~\\ \\  \\ /:/  \\:\\  \\ /::\\~\\:\\  \\ /::\\  \\   /::\\__\\:/|:| |__  /::\\~\\:\\  \\ \\r\n"
        + "    /\\ \\:\\ \\ \\__\\_\\/__/:/__/ \\:\\__\\_\\:/\\:\\:\\  \\_\\:/:\\:\\  \\ /:/\\__/:|_ /\\__\\_\\:/__/_\\:\\:\\__\\ \\r\n"
        + "    \\:\\ \\:\\ \\/__/\\_\\:\\  \\ /:/  /\\/_|::\\:/:/  /\\/__\\_\\/:/  / \\/_|::|./\\:\\  \\ /\\ \\/__/ \\r\n"
        + "     \\:\\ \\:\\__\\  \\:\\  /:/  /   |:|::/:/  /   \\::/__/      |:/:/  / \\:\\  \\:\\__\\ \\r\n"
        + "      \\:\\/:/  /   \\:\\/:/  /   |:|\\/__/__/    \\:\\__\\      |::/  /   \\:\\/:/  / \\r\n"
        + "       \\::/  /     \\::/  /    |:| |         \\/__/       /:/  /     \\::/  / \\r\n"
        + "        \\/__/       \\/__/      \\|__|                     \\/__/       \\/__/  ";
```

```java
        try {
            for (int i = 0; i < sorting.length(); i++)
            {
                System.out.print(sorting.charAt(i));
                TimeUnit.MILLISECONDS.sleep(1);
            }
        }
        catch (Exception e){
        }
        Tools.Space();
        System.out.print("  +");
        Tools.PrintLine(11, 0);
        Tools.Space();
    }
    static void Menu()
    {
        System.out.print("\tMenu\n\t\t1. Bubble\n"
                + "\t\t2. Insertion\n\t\t3. Selection\n"
                + "\t\t4. Exit\n\tEnter Option[1..4] ");
    }
    static void End()
    {
        System.out.print("\t\tThank you po for checking my program ^_^\n"
                + "\t\t\t-Marasigan, Vem Aiensi A.\n\t+");
        Tools.PrintLine(6, 0);
        System.out.println("\n\t  Special Thanks also to: patorjk.com for the ASCII art"
                + "\n\t\t  https://patorjk.com/software/taag/");
        System.exit(0);
    }
}
```

SCREENSHOTS
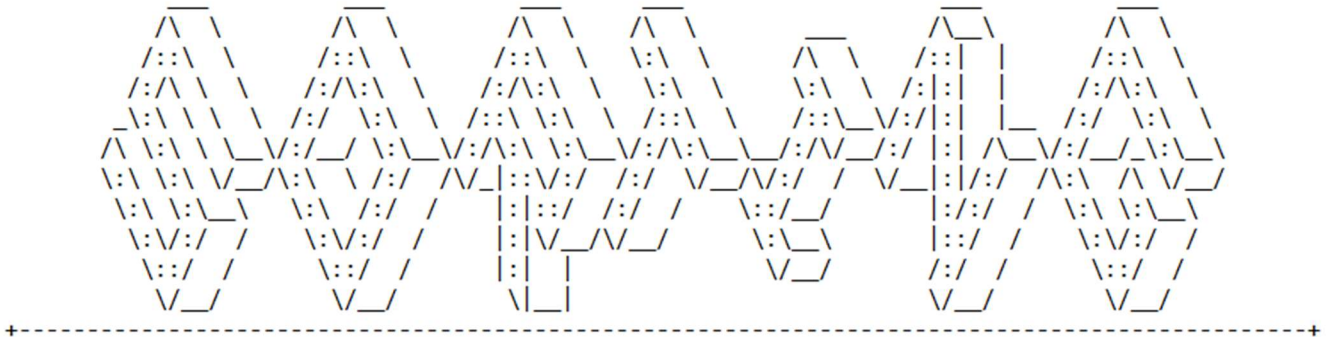
Sample Input 1	Sample Input 2
Enter the size of an array: 5	Enter the size of an array: 7
Enter 5 values: 3 1 5 4 2	Enter 7 values: 70 6 23 4 2 1 15

BUBBLE

```
Enter Option[1..4] 1

+--[ BUBBLE ]-----------------------------------+
|        3        1        5        4        2   |
+--1st  PASS------------------------------------+
|        1        3        5        4        2   |
|        1        3        5        4        2   |
|        1        3        4        5        2   |
|        1        3        4        2        5   |
+--2nd  PASS------------------------------------+
|        1        3        4        2        5   |
|        1        3        4        2        5   |
|        1        3        2        4        5   |
|        1        3        2        4        5   |
+--3rd  PASS------------------------------------+
|        1        3        2        4        5   |
|        1        2        3        4        5   |
|        1        2        3        4        5   |
|        1        2        3        4        5   |
+-----------------------------------------------+
+--SORTED ARRAY---------------------------------+
|        1        2        3        4        5   |
+-----------------------------------------------+
```

```
Enter Option[1..4] 1

+--[ BUBBLE ]-----------------------------------------------+
|      70      6      23      4      2      1      15        |
+--1st  PASS------------------------------------------------+
|       6     70      23      4      2      1      15        |
|       6     23      70      4      2      1      15        |
|       6     23       4     70      2      1      15        |
|       6     23       4      2     70      1      15        |
|       6     23       4      2      1     70      15        |
|       6     23       4      2      1     15      70        |
+--2nd  PASS------------------------------------------------+
|       6     23       4      2      1     15      70        |
|       6      4      23      2      1     15      70        |
|       6      4       2     23      1     15      70        |
|       6      4       2      1     23     15      70        |
|       6      4       2      1     15     23      70        |
|       6      4       2      1     15     23      70        |
+--3rd  PASS------------------------------------------------+
|       4      6       2      1     15     23      70        |
|       4      2       6      1     15     23      70        |
|       4      2       1      6     15     23      70        |
|       4      2       1      6     15     23      70        |
|       4      2       1      6     15     23      70        |
|       4      2       1      6     15     23      70        |
+--4th  PASS------------------------------------------------+
|       2      4       1      6     15     23      70        |
|       2      1       4      6     15     23      70        |
|       2      1       4      6     15     23      70        |
|       2      1       4      6     15     23      70        |
|       2      1       4      6     15     23      70        |
|       2      1       4      6     15     23      70        |
+--5th  PASS------------------------------------------------+
|       1      2       4      6     15     23      70        |
|       1      2       4      6     15     23      70        |
|       1      2       4      6     15     23      70        |
|       1      2       4      6     15     23      70        |
|       1      2       4      6     15     23      70        |
|       1      2       4      6     15     23      70        |
+-----------------------------------------------------------+
+--SORTED ARRAY--------------------------------------------+
|       1      2       4      6     15     23      70        |
+-----------------------------------------------------------+
```

INSERTION

```
Enter Option[1..4] 2

+--[ INSERTION ]--------------------------------+
|        3        1        5        4        2   |
+--1st  PASS------------------------------------+
|        1        3        5        4        2   |
+--2nd  PASS------------------------------------+
|        1        3        5        4        2   |
+--3rd  PASS------------------------------------+
|        1        3        4        5        2   |
+--4th  PASS------------------------------------+
|        1        2        3        4        5   |
+-----------------------------------------------+
+--SORTED ARRAY---------------------------------+
|        1        2        3        4        5   |
+-----------------------------------------------+
```

```
Enter Option[1..4] 2

+--[ INSERTION ]--------------------------------------------+
|      70      6      23      4      2      1      15        |
+--1st  PASS------------------------------------------------+
|       6     70      23      4      2      1      15        |
+--2nd  PASS------------------------------------------------+
|       6     23      70      4      2      1      15        |
+--3rd  PASS------------------------------------------------+
|       4      6      23     70      2      1      15        |
+--4th  PASS------------------------------------------------+
|       2      4       6     23     70      1      15        |
+--5th  PASS------------------------------------------------+
|       1      2       4      6     23     70      15        |
+--6th  PASS------------------------------------------------+
|       1      2       4      6     15     23      70        |
+-----------------------------------------------------------+
+--SORTED ARRAY--------------------------------------------+
|       1      2       4      6     15     23      70        |
+-----------------------------------------------------------+
```

SELECTION

```
Menu
        1. Bubble
        2. Insertion
        3. Selection
        4. Exit
Enter Option[1..4] 3

+--[ SELECTION ]-------------------------------+
|        3        1        5        4        2        |
+--1st  PASS-----------------------------------+
|        1        3        5        4        2        |
+--2nd  PASS-----------------------------------+
|        1        2        5        4        3        |
+--3rd  PASS-----------------------------------+
|        1        2        3        4        5        |
+----------------------------------------------+
+---SORTED ARRAY-------------------------------+
|        1        2        3        4        5        |
+----------------------------------------------+
```

```
Menu
        1. Bubble
        2. Insertion
        3. Selection
        4. Exit
Enter Option[1..4] 3

+--[ SELECTION ]-----------------------------------------------+
|       70       6       23       4       2       1       15       |
+--1st  PASS---------------------------------------------------+
|        1       6       23       4       2       70       15       |
+--2nd  PASS---------------------------------------------------+
|        1       2       23       4       6       70       15       |
+--3rd  PASS---------------------------------------------------+
|        1       2        4       23      6       70       15       |
+--4th  PASS---------------------------------------------------+
|        1       2        4        6      23       70       15       |
+--5th  PASS---------------------------------------------------+
|        1       2        4        6      15       70       23       |
+--6th  PASS---------------------------------------------------+
|        1       2        4        6      15       23       70       |
+-------------------------------------------------------------+
+--SORTED ARRAY-----------------------------------------------+
|        1       2        4        6      15       23       70       |
+-------------------------------------------------------------+
```

EXIT

```
        Enter Option[1..4] 4

        Thank you po for checking my program ^_^
                -Marasigan, Vem Aiensi A.
+------------------------------------------------------+
        Special Thanks also to: patorjk.com for the ASCII art
                https://patorjk.com/software/taag/
```

VIDEOS IN RUNNING

Drive Link: https://drive.google.com/file/d/1e4VhkhnvK_cSLtXVDsE-Eok8KzBqJQk-/view?usp=sharing