2.

a)

Bubble Sort

```
package Lab_2;

import java.util.Random;

public class ArrayBub {

      private long[] a;                 // ref to array a
         private int nElems;               // number of data items
      //--------------------------------------------------------------
         public ArrayBub(int max)          // constructor
            {
            a = new long[max];                 // create the array
            nElems = 0;                        // no items yet
            }
      //--------------------------------------------------------------
         public void insert(long value)    // put element into array
            {
            a[nElems] = value;             // insert it
            nElems++;                      // increment size
            }
      //--------------------------------------------------------------
         public void display()             // displays array contents
            {
            for(int j=0; j<nElems; j++)       // for each element,
               System.out.print(a[j] + " ");  // display it
            System.out.println("");
            }
      //--------------------------------------------------------------
         public void bubbleSort()
            {
            int out, in;

            for(out=nElems-1; out>1; out--)    // outer loop (backward)
               for(in=0; in<out; in++)         // inner loop (forward)
                  if( a[in] > a[in+1] )        // out of order?
                     swap(in, in+1);           // swap them
            }  // end bubbleSort()
      //--------------------------------------------------------------
         private void swap(int one, int two)
            {
            long temp = a[one];
            a[one] = a[two];
            a[two] = temp;
            }
      //--------------------------------------------------------------
         public void randomInit(int numElements) {
               Random aRandom = new Random();
               nElems = numElements;
               for(int i = 0; i<nElems;i++) {
                     a[i] = aRandom.nextLong() % 100000000;
               }
         }
}
```

```
package Lab_2;

public class BubbleSortApp {
```

```
      public static void main(String[] args)
    {

    int maxSize = 10000;            // array size
    ArrayBub arr;                 // reference to array
    arr = new ArrayBub(maxSize);  // create the array
    arr.randomInit(maxSize);    // Create random values
//    for(int i=0; i<maxSize;i++) { // Already-sorted order
//          arr.insert(i);
//    }
//    for(int i=maxSize; i>0;i--) { // Inverse order
//          arr.insert(i);
//    }
//
//    arr.display();                // display items
      long start = TimeUtils.now();
    arr.bubbleSort();             // bubble sort them
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    //arr.display();                // display them again
    }  // end main()
 }  // end class BubbleSortApp
```

Selection Sort

```
package Lab_2;

import java.util.Random;

public class ArraySel {

      private long[] a;                 // ref to array a
         private int nElems;               // number of data items
      //--------------------------------------------------------------
         public ArraySel(int max)          // constructor
            {
            a = new long[max];                 // create the array
            nElems = 0;                        // no items yet
            }
      //--------------------------------------------------------------
         public void insert(long value)    // put element into array
            {
            a[nElems] = value;             // insert it
            nElems++;                      // increment size
            }
      //--------------------------------------------------------------
         public void display()             // displays array contents
            {
            for(int j=0; j<nElems; j++)       // for each element,
               System.out.print(a[j] + " ");  // display it
            System.out.println("");
            }
      //--------------------------------------------------------------
         public void selectionSort()
            {
            int out, in, min;

            for(out=0; out<nElems-1; out++)    // outer loop
               {
               min = out;                      // minimum
```

```
                for(in=out+1; in<nElems; in++) // inner loop
                   if(a[in] < a[min] )         // if min greater,
                       min = in;               // we have a new min
                swap(out, min);                // swap them
                }  // end for(out)
            }  // end selectionSort()
      //--------------------------------------------------------------
         private void swap(int one, int two)
            {
            long temp = a[one];
            a[one] = a[two];
            a[two] = temp;
            }
       //--------------------------------------------------------------
         public void randomInit(int numElements) {
               Random aRandom = new Random();
               nElems = numElements;
               for(int i = 0; i<nElems;i++) {
                     a[i] = aRandom.nextLong() % 100000000;
               }
         }
}
```

```
package Lab_2;

public class SelectSortApp {

      public static void main(String[] args)
    {


        int maxSize = 10000;            // array size
        ArraySel arr;                 // reference to array
        arr = new ArraySel(maxSize);  // create the array
        arr.randomInit(maxSize);          // Create random values
//      for(int i=maxSize; i>0;i--) { // Already-sorted order
//            arr.insert(i);
//      }
//          for(int i=maxSize; i>0;i--) { // Inverse order
//          arr.insert(i);
//    }

    //arr.display();                // display items
      long start = TimeUtils.now();
    arr.selectionSort();          // selection-sort them

    //arr.display();                // display them again
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    }
}
```

Insertion Sort

```
package Lab_2;

import java.util.Random;

public class ArrayIns {
```

```
      private long[] a;                 // ref to array a
         private int nElems;               // number of data items
      //--------------------------------------------------------------
         public ArrayIns(int max)          // constructor
            {
            a = new long[max];                 // create the array
            nElems = 0;                        // no items yet
            }
      //--------------------------------------------------------------
         public void insert(long value)    // put element into array
            {
            a[nElems] = value;             // insert it
            nElems++;                      // increment size
            }
      //--------------------------------------------------------------
         public void display()             // displays array contents
            {
            for(int j=0; j<nElems; j++)       // for each element,
               System.out.print(a[j] + " ");  // display it
            System.out.println("");
            }
      //--------------------------------------------------------------
         public void insertionSort()
            {
            int in, out;

            for(out=1; out<nElems; out++)     // out is dividing line
               {
               long temp = a[out];            // remove marked item
               in = out;                      // start shifts at out
               while(in>0 && a[in-1] >= temp) // until one is smaller,
                  {
                  a[in] = a[in-1];            // shift item to right
                  --in;                       // go left one position
                  }
               a[in] = temp;                  // insert marked item
               }  // end for
            }  // end insertionSort()
       //--------------------------------------------------------------
         public void randomInit(int numElements) {
               Random aRandom = new Random();
               nElems = numElements;
               for(int i = 0; i<nElems;i++) {
                     a[i] = aRandom.nextLong() % 100000000;
               }
         }
}
```

```
package Lab_2;

public class InsertSortApp {

      public static void main(String[] args)
    {

    int maxSize = 10000;            // array size
    ArrayIns arr;                 // reference to array
    arr = new ArrayIns(maxSize);  // create the array
    arr.randomInit(maxSize);      // Create random values
//    for(int i=maxSize; i>0;i--) { // Already-sorted order
//          arr.insert(i);
//    }
```

```
//    for(int i=maxSize; i>0;i--) { // Inverse order
//          arr.insert(i);
//    }

//    arr.display();                // display items
      long start = TimeUtils.now();
    arr.insertionSort();          // insertion-sort them
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    //arr.display();                // display them again
    }  // end main()
}
```

Time Utils

```
package Lab_2;

import java.util.Calendar;

public class TimeUtils {

      public static long now() {
            Calendar cal = Calendar.getInstance();
            java.util.Date currentDate = cal.getTime();
            return currentDate.getTime();       }
}
```

b)

Inverse Order

BubbleSort

```
package Lab_2;

public class BubbleSortApp {

      public static void main(String[] args)
    {

    int maxSize = 10000;            // array size
    ArrayBub arr;                 // reference to array
    arr = new ArrayBub(maxSize);  // create the array
    //arr.randomInit(maxSize);    // Create random values
//    for(int i=0; i<maxSize;i++) { // Already-sorted order
//          arr.insert(i);
//    }
      for(int i=maxSize; i>0;i--) { // Inverse order
            arr.insert(i);
      }
//
//    arr.display();                // display items
      long start = TimeUtils.now();
    arr.bubbleSort();             // bubble sort them
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    //arr.display();                // display them again
    }  // end main()
 }  // end class BubbleSortApp
```

Selection Sort

```
package Lab_2;

public class SelectSortApp {

      public static void main(String[] args)
    {


        int maxSize = 10000;            // array size
        ArraySel arr;                 // reference to array
        arr = new ArraySel(maxSize);  // create the array
        //arr.randomInit(maxSize);        // Create random values
//      for(int i=maxSize; i>0;i--) { // Already-sorted order
//            arr.insert(i);
//      }
            for(int i=maxSize; i>0;i--) { // Inverse order
            arr.insert(i);
      }

    //arr.display();                // display items
      long start = TimeUtils.now();
    arr.selectionSort();          // selection-sort them

    //arr.display();                // display them again
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    }
}
```

Insertion Sort

```
package Lab_2;

public class InsertSortApp {

      public static void main(String[] args)
    {

    int maxSize = 10000;            // array size
    ArrayIns arr;                 // reference to array
    arr = new ArrayIns(maxSize);  // create the array
    //arr.randomInit(maxSize);      // Create random values
//    for(int i=maxSize; i>0;i--) { // Already-sorted order
//          arr.insert(i);
//    }
      for(int i=maxSize; i>0;i--) { // Inverse order
            arr.insert(i);
      }

//    arr.display();                // display items
      long start = TimeUtils.now();
    arr.insertionSort();          // insertion-sort them
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    //arr.display();                // display them again
    }  // end main()
```

```
}
```

Already-sorted order

Bubble Sort

```
package Lab_2;

public class BubbleSortApp {

      public static void main(String[] args)
    {

    int maxSize = 10000;            // array size
    ArrayBub arr;                 // reference to array
    arr = new ArrayBub(maxSize);  // create the array
    //arr.randomInit(maxSize);    // Create random values
      for(int i=0; i<maxSize;i++) { // Already-sorted order
            arr.insert(i);
      }
//    for(int i=maxSize; i>0;i--) { // Inverse order
//          arr.insert(i);
//    }
//
//    arr.display();                // display items
      long start = TimeUtils.now();
    arr.bubbleSort();             // bubble sort them
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    //arr.display();                // display them again
    }  // end main()
 }  // end class BubbleSortApp
```

Selection Sort

```
package Lab_2;

public class SelectSortApp {

      public static void main(String[] args)
    {


        int maxSize = 10000;            // array size
        ArraySel arr;                 // reference to array
        arr = new ArraySel(maxSize);  // create the array
        //arr.randomInit(maxSize);        // Create random values
        for(int i=maxSize; i>0;i--) { // Already-sorted order
              arr.insert(i);
        }
//      for(int i=maxSize; i>0;i--) { // Inverse order
//          arr.insert(i);
//      }

    //arr.display();                // display items
      long start = TimeUtils.now();
    arr.selectionSort();          // selection-sort them

    //arr.display();                // display them again
```

```
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    }
}
```

Insertion Sort

```
package Lab_2;

public class InsertSortApp {

      public static void main(String[] args)
    {

    int maxSize = 10000;            // array size
    ArrayIns arr;                 // reference to array
    arr = new ArrayIns(maxSize);  // create the array
    //arr.randomInit(maxSize);      // Create random values
      for(int i=maxSize; i>0;i--) { // Already-sorted order
            arr.insert(i);
      }
//    for(int i=maxSize; i>0;i--) { // Inverse order
//          arr.insert(i);
//    }

//    arr.display();                // display items
      long start = TimeUtils.now();
    arr.insertionSort();          // insertion-sort them
    long end = TimeUtils.now();
    long time = end - start;
    System.out.println("Time elapsed: "+time);
    //arr.display();                // display them again
    }  // end main()
}
```