```
package com.Jaynish;

public class Main {

  public static void main(String[] args) {

  }

  public static void Problem4A(int[] a, int arraySize) {

      // minima array
      int[] b = new int[arraySize];
      b = a.clone();
      int i = 0;

      while (i < arraySize && i + 1 < arraySize) {    // O(N)
          if (a[i] > a[i + 1]) {
              int temp = a[i];
              a[i] = a[i + 1];
              a[i + 1] = temp;
              i += 2;
          } else {
              i++;
          }
      }
      System.out.println("Minima:");
      for (i = 0; i < arraySize; i++) {  // O(N)
          System.out.print(a[i] + " ");
      }


      int k = 0;

      while (k < b.length && k + 1 < b.length) {   //O(N)
          if (b[k] < b[k + 1]) {
              int temp = b[k];
              b[k] = b[k + 1];
              b[k + 1] = temp;
              k += 2;
          } else {
              k++;
          }
      }

      System.out.println("\n");
      System.out.print("Maxima:");

      for (k = 0; k < b.length; k++) {   //O(N)
          System.out.print(b[k] + " ");
      }
```

iii)

The loops highlighted in orange have time complexity of of O(N). The while loops that are show are only loops N times.

iv)

A linear time solution does exist for this problem. As state above the time complexity of the code snippet is O(N)