CSC 225 - Assignment 3 Analysis Report
Alex Holland

```
public static int CountInversions(int[] A){

  int invCount = 0;                        //Track the number of inversions in the array.
  int temp;                                //Temporary array value holder.

  for (int i = 0; i < A.length - 1; i++){  //1. Loop while i is one less than the size of the array.
     if (A[i] > A[i + 1]) {                //2. Determine if element 1 and element2 should be swapped.
        temp = A[i];                       //3. Hold the element 1 value in temp.
        A[i] = A[i + 1];                   //4. Set element 1 to be element 2.
        A[i + 1] = temp;                   //5. Set element 2 to be temp which holds element 1.
        invCount++;                        //6. Increase the inversion count.
        if (i >= 1){                       //7. If two elements are swapped go back two indexes and
           i -= 2;                         // determine If the new index and next element should also be
        }                                  // swapped.
     }
  }
  return invCount;                         //Return the number of inversion in the array.
}
```

The loop executes A.length - 2 times.

The implemented algorithm is $O(n + k)$ on an array with n elements and k inversions, resulting in a $O(n)$ algorithm when $k \in O(n)$.

Counting operations of worst case time running time T(n)

Primitive Operations:

- Assignments (A)
- Comparisons (C)
- Array indexing (I)
- Add, subtract (S)

$CountInversions(int[]\ A)$:
  $Input:\ An\ array\ of\ elements.$
  $Output:\ The\ number\ of\ counted\ inversions.$

$invCount \leftarrow 0$	$1A$
$temp \leftarrow 0$	$1A$
$for\ i \leftarrow 0\ to\ A.length - 2\ do$	$1A + (n - 1)(1C + 1A + 1S) + 1C$
$\quad if\ A[i] > A[i + 1]\ then$	$(n - 1)(1C + 2I + 1S)$
$\quad\quad swap(i,\ i + 1)$	$(n - 1)(4A + 4I + 3S)$
$\quad end$	
$\quad if\ i \geq 1$	$(n - 1)(1C)$
$\quad\quad i \leftarrow i - 2$	$(n - 1)(1A + 1S)$
$\quad end$	
end	

$return\ invCount$ $\qquad (n-1)(1A)$

$T(n) = 1 + 1 + 1 + 3(n-1) + 1 + 3(n-1) + 11(n-1) + (n-1) + 2(n-1) + 1(n-1)$
$T(n) = 4 + 3n - 3 + 3n - 3 + 11n - 11 + n - 1 + 2n - 2 + n - 1$
$T(n) = 21n - 17$

We can see that $k \geq n$.