

# CPSC 331

## Tutorial4

Department of Computer Science  
University of Calgary

# Running Time

- ❖ Number of primitive operations or “steps” (programming language statements) used
- ❖ Worst Case Analysis
- ❖ Average Case Analysis
- ❖ Best Case Analysis

As part of the previous tutorial exercise, you tested and debugged a program that would be used to determine whether the entries of a given array are distinct. A corrected version of this program is given.

```
public static boolean distinctEntries ( int[] A ) {
```

??? for (int i=1; i <= A.length; i++ ) { For (int i=1;i<A.lenght;i++)

**Loop Invariant**

```
* Loop Invariant:
* a) i is an integer such that 1 <= i < A.length
* b) A[r] is not equal to A[s] for all integers r
*    and s such that 0 <= r < s < i
```

**Loop variant**

```
* Loop Variant: A.length - i
*/
```

??? for (int j=1; j <= i; j++ ) { For (int j=0; j<i;j++)

**Loop Invariant**

```
* Loop Invariant:
* a) i and j are integers such that
*    0 <= j < i < A.length
* b) A[r] is not equal to A[s] for all
*    integers r and s such that
*    0 <= r < s < i
* c) A[r] is not equal to A[i] for every
*    integer r such that
*    0 <= r < j
```

**Loop variant**

```
* Loop Variant: i-j
*/
```

```
    if (A[j] = A[i]) {
        return false;
    };
};
return true;
}
```

```

public static boolean distinctEntries ( int[] A ) {

    for (int i=1; i < A.length; i++ ) {
        /*
        * Loop Invariant:
        * a) i is an integer such that  $1 \leq i < A.length$ 
        * b) A[r] is not equal to A[s] for all integers r
        *    and s such that  $0 \leq r < s < i$ 
        * Loop Variant:  $A.length - i$ 
        */
        for (int j=0; j < i; j++ ) {
            /*
            * Loop Invariant:
            * a) i and j are integers such that
            *     $0 \leq j < i < A.length$ 
            * b) A[r] is not equal to A[s] for all
            *    integers r and s such that
            *     $0 \leq r < s < i$ 
            * c) A[r] is not equal to A[i] for every
            *    integer r such that
            *     $0 \leq r < j$ 
            * Loop Variant:  $i-j$ 
            */
            if (A[j] == A[i]) {
                return false;
            };
        };
    };
    return true;
}

```

# Questions

a)

Analyze the *worst case running time* of this program.

Answer:

$O(n^2)$

# Questions

**b)** Try to analyze the *best case running time* of this program.

Answer:

$O(1)$

# Questions

- ❖ c) What, if anything, can be said about the *average* (or “*expected*”) running time of this program, based on what you have discovered?

Answer:

$$n/2 * n/2 \Rightarrow O(n^2)$$