

# CS2013: Programación III

## Laboratorio: Complejidad Algorítmica I

### Laboratorio 7A

José Chávez

# Agenda

- Invariante de bucle
- Prueba de correctitud

# Ejercicios

## Ejercicio 1:

El siguiente algoritmo calcula el promedio de una secuencia de números  $V$ .  
Utilizando el invariante de bucle, demostrar la correctitud del algoritmo:

```
promedio(V):  
1. n = V.size()  
2. s = 0  
3. for i in 0, ..., n-1  
4.     s = s + V[i]  
5. return s / n
```

## Ejercicio 2:

El siguiente algoritmo calcula el máximo elemento de una secuencia de números  $V$ . Utilizando el invariante de bucle, demostrar la correctitud del algoritmo:

```
max_elemento(V):  
1. max = V[0]  
2. for i in 1, ..., n-1  
3.     if V[i] > max  
4.         max = V[i]  
5. return max
```

## Ejercicio 3:

El siguiente algoritmo retorna el índice en el cual se encuentra el elemento  $x$  en una secuencia de números  $V$ . Si  $x$  no se encuentra en  $V$ , entonces retorna NULL. Utilizando el invariante de bucle, demostrar la correctitud del algoritmo:

```
linear_search(V, x):  
1. n = size(V)  
2. for i in 0, ..., n-1  
3.     if V[i] == x  
4.         return i  
5. return NULL
```

## Ejercicio 4:

El siguiente algoritmo calcula la multiplicación de dos números naturales. Utilizando el invariante de bucle, demostrar la correctitud del algoritmo:

`multiplicacion(a, b):`

```
1. x, y = a, b
2. total = 0
3. while x > 0
4.     if x % 2 == 1
5.         total = total + y
6.     x = x // 2
7.     y = y * 2
8. return total
```

## Ejercicio 5:

Demostrar la correctitud del algoritmo *Bubble Sort*

```
bubble_sort(A):  
1. n = size(A)  
2. for i=0 to n-1  
3.     for j=n-1 down to i+1  
4.         if A[j-1] > A[j]  
5.             swap(A[j-1], A[j])
```



# Resumen

En esta sesión se trataron los tópicos siguientes:

- El invariante de bucle
- Prueba de correctitud de algoritmos

