

ALGORITMOS Y ESTRUCTURA DE DATOS I

MINI INTRO AL TALLER DE PROGRAMACIÓN CON INVARIANTES

23 de Abril de 2018

Sintaxis de un ciclo:

```
while(B) {  
    // cuerpo del ciclo  
}
```

- ▶ El ciclo se repite continuamente mientras la guarda B se cumpla. Cada repetición es una iteración.
- ▶ El ciclo termina cuando no se cumpla la guarda.
- ▶ Al salir, en caso de que el ciclo terminara, el estado resultante es el mismo que el del final de la última iteración.

REPASO - TEOREMA DEL INVARIANTE

Sea P_C la precondition del ciclo, Q_C la postcondición, B la guarda e I un invariante del ciclo. Si se cumple:

1. $P_C \Rightarrow I$,
2. $\{I \wedge B\}$ cuerpo del ciclo $\{I\}$,
3. $I \wedge \neg B \Rightarrow Q_C$,

entonces el ciclo es **parcialmente** correcto (si termina, termina en Q_C).

```
// Vale I
while(B){
    // Vale I && B
    Cuerpo del ciclo
    // Vale I
}
// Vale Qc
```

EJEMPLO

```
bool hayMayorACero(vector<int> v) {  
    bool encuentre = false;  
    int i = 0;  
    int n = v.size();  
    while(i<n) {  
        encuentre = encuentre || v[i] > 0;  
        i = i+1;  
    }  
    return encuentre;  
}
```

► Sea $v = \{-1, -2, 3, -3, 4\}$

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
-----------	---	----------

FINAL DE ITERACIÓN

Iteración	i	encontre
-----------	---	----------

EJEMPLO

```
bool hayMayorACero(vector<int> v) {  
    bool encuentre = false;  
    int i = 0;  
    int n = v.size();  
    while(i<n) {  
        encuentre = encuentre || v[i] > 0;  
        i = i+1;  
    }  
    return encuentre;  
}
```

► Sea $v = \{-1, -2, 3, -3, 4\}$

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false

EJEMPLO

```
bool hayMayorACero(vector<int> v) {  
    bool encuentre = false;  
    int i = 0;  
    int n = v.size();  
    while(i<n) {  
        encuentre = encuentre || v[i] > 0;  
        i = i+1;  
    }  
    return encuentre;  
}
```

► Sea $v = \{-1, -2, 3, -3, 4\}$

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false

EJEMPLO

```
bool hayMayorACero(vector<int> v) {  
    bool encuentre = false;  
    int i = 0;  
    int n = v.size();  
    while(i<n) {  
        encuentre = encuentre || v[i] > 0;  
        i = i+1;  
    }  
    return encuentre;  
}
```

► Sea $v = \{-1, -2, 3, -3, 4\}$

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true

EJEMPLO

```
bool hayMayorACero(vector<int> v) {  
    bool encuentre = false;  
    int i = 0;  
    int n = v.size();  
    while(i<n) {  
        encuentre = encuentre || v[i] > 0;  
        i = i+1;  
    }  
    return encuentre;  
}
```

► Sea $v = \{-1, -2, 3, -3, 4\}$

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true

EJEMPLO

```
bool hayMayorACero(vector<int> v) {  
    bool encuentre = false;  
    int i = 0;  
    int n = v.size();  
    while(i<n) {  
        encuentre = encuentre || v[i] > 0;  
        i = i+1;  
    }  
    return encuentre;  
}
```

► Sea $v = \{-1, -2, 3, -3, 4\}$

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

EJEMPLO

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

ES UN INVARIANTE?

► $I \equiv i \leq n$?

EJEMPLO

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

ES UN INVARIANTE?

► $I \equiv i \leq n$? 👍

► $I \equiv i \leq n \wedge \text{encontre} = \text{true} \vee \text{false} ?$

EJEMPLO

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

ES UN INVARIANTE?

- ▶ $I \equiv i \leq n$? 👍
- ▶ $I \equiv i \leq n \wedge \text{encontre} = \text{true} \vee \text{false}$? 👍
- ▶ $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k \leq i \wedge v[k] > 0$?

EJEMPLO

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

ES UN INVARIANTE?

► $I \equiv i \leq n$? 👍

► $I \equiv i \leq n \wedge \text{encontre} = \text{true} \vee \text{false}$? 👍

► $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k \leq i \wedge v[k] > 0$? 😞 👎

EJEMPLO

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

ES UN INVARIANTE?

- ▶ $I \equiv i \leq n$? 👍
- ▶ $I \equiv i \leq n \wedge \text{encontre} = \text{true} \vee \text{false}$? 👍
- ▶ $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k \leq i \wedge v[k] > 0$? 😟 👎
- ▶ $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k < i \wedge v[k] > 0$?

EJEMPLO

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

ES UN INVARIANTE?

- ▶ $I \equiv i \leq n$? 👍
- ▶ $I \equiv i \leq n \wedge \text{encontre} = \text{true} \vee \text{false}$? 👍
- ▶ $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k \leq i \wedge v[k] > 0$? 😟 👎
- ▶ $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k < i \wedge v[k] > 0$? 🙌 💪

EJEMPLO

PRINCIPIO DE ITERACIÓN

Iteración	i	encontre
1	0	false
2	1	false
3	2	false
4	3	true
5	4	true

FINAL DE ITERACIÓN

Iteración	i	encontre
1	1	false
2	2	false
3	3	true
4	4	true
5	5	true

ES UN INVARIANTE?

► $I \equiv i \leq n$? 👍

► $I \equiv i \leq n \wedge \text{encontre} = \text{true} \vee \text{false}$? 👍

► $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k \leq i \wedge v[k] > 0$? 😬 🙅

► $I \equiv 0 \leq i \leq n \wedge \text{encontre} = (\exists k : \mathbb{Z}) 0 \leq k < i \wedge v[k] > 0$? 🙅 💪



No todos nos van a servir para poder demostrar la correctitud parcial del ciclo! (en particular $I \wedge \neg B \Rightarrow Q_c$)

EJERCICIO

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 3 \leq i \leq n + 1 \wedge suma = \sum_{k=3}^{i-1} \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

EJERCICIO

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 3 \leq i \leq n + 1 \wedge suma = \sum_{k=3}^{i-1} \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

```
int suma = 0;
int i = 3;
while(i <= n) {
    if (esPrimo(i)){
        suma = suma + i;
    }
    i++;
}
return suma;
```

VARIANTE 1

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 3 \leq i \leq n + 2 \wedge i \bmod 2 = 1 \wedge$$

$$suma = \sum_{k=3}^{i-2} \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

VARIANTE 1

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 3 \leq i \leq n + 2 \wedge i \bmod 2 = 1 \wedge$$

$$suma = \sum_{k=3}^{i-2} \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

```
int suma = 0;
int i = 3;
while(i <= n) {
    if (esPrimo(i)){
        suma = suma + i;
    }
    i = i + 2;
}
return suma;
```

VARIANTE 2

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 1 \leq i \leq n \wedge i \bmod 2 = 1 \wedge$$
$$suma = \sum_{k=i+2}^n \text{ if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

VARIANTE 2

Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive) respetando el siguiente invariante:

$$I \equiv 1 \leq i \leq n \wedge i \bmod 2 = 1 \wedge$$

$$suma = \sum_{k=i+2}^n \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

```
int suma = 0;
int i = n;
if(i % 2 == 0 )
    i--;
while(i > 2) {
    if (esPrimo(i)){
        suma = suma + i;
    }
    i -= 2;
}
return suma;
```