

Algoritmo(int n){

¿Cual es la forma de los estados?

¿Cual es el estado inicial?

¿Cómo es la transición de datos?

¿Cómo es el estado final?

¿Cómo es la invariante de ciclo?

```

i = 0
s = 3
while(i <= n){
    j = 0
    p = 4
    while(j <= 2n){
        p += 2
        j += 1
    }
    s += 2p
    i += 2
}

```

1. Considerar el ciclo externo, reduciendo el interno a una expresión.  $p = ?$ .

2. Considerar el ciclo interno de forma independiente.

$p = 4$   
 $j = 0$   
 $4+2$   
 $4+2+2$   
 $4+2 \times 3$   
 $4+2 \times 4$   
 $4 \times 2 \times 5$   
 $4+2 \times 2n$   
 $4+4n$   
 $4n+6$   
 $4+4n+2$   
 $2n+1$

$$p = 4n + 6$$

$$s = 8n + 12$$

0, 2, 4, 6, ..., n, n+2

Ciclo externo

Forma estado  $C = (i, s)$

Estado inicial  $(0, 3)$

Transformación  $(i, s) \rightarrow (i+2, s+8n+12)$

$(0, 3) \rightarrow (2, 3+8n+12) \rightarrow (4, 3+2(8n+12)) \rightarrow (6, 3+3(8n+12))$   
 $\rightarrow (8, 3+4(8n+12))$

$$(i, 3 + \frac{i}{2}(8n+12))$$

Invariante de ciclo

Estado final

$$(n+2, 3 + \frac{n+2}{2}(8n+12))$$

Verificación

- 1) Invariante  $I_0$   $(0, 3) \rightsquigarrow (0, 3) \checkmark$
- 2) Invariante  $I_T$   $(i, s) \rightarrow (i+2, s+8n+12)$

$(i, 3 + \frac{i}{2}(8n+12))$   
 $i = i+2$   
 $(i+2, \frac{i+2}{2}(8n+12))$  *Guayando*  
 $(i+2, \frac{i}{2}(8n+12) + (8n+12))$   
 $(i+2, (\frac{i}{2}+1)(8n+12))$   
 $(i+2, \frac{i+2}{2}(8n+12))$  *Base inductive*

Estado final  $i = n+2$

(Trivial)

$$(n+2, \frac{n+2}{2}(8n+12))$$

Algoritmo(int n){

i = 0

s = 3

while(i <= n){

j = 0

p = 4

while(j <= 2n){

p += 2

j += 1

s += 2p

i += 2

}

}

Forma Estado  $(j, p)$

Estado inicial  $(0, 4)$

Transformación

$(j, p) \rightarrow (j+1, p+2)$

$(0, 4) \rightarrow (1, 4+2) \rightarrow (2, 4+(2)2) \rightarrow (3, 4+(3)2) \rightarrow (4, 4(4)2)$

$(j, 4+j(2))$   
Invariante

Estado final

$0-1, 2, 3, 4, 5, 6, \dots, 2n, 2n+1)$

$4n+6$

Estado final

$(2n+1, 4+4n+2)$

$(2n+1, 4n+6)$

Invariante

$j = j+1$

$(j+1, 4n+2(j+1))$   
 $(j+1, 4n+2j+2)$

Evaluar la invariante

$(j, 4n+2j) \rightarrow (j+1, 4n+2j+2)$

Aplicar la transformación

Estado inicial  $\Rightarrow j=0$   $(0, 4)$

Estado final  $\Rightarrow j=2n+1$   $(2n+1, 4n+6)$

```

BS (int A[], int N){
    int i, j, aux;
    for (i=1; i < N; i++){
        for (j=N; j > i; j--){
            if (A[j] < A[i]){
                aux=A[j];
                A[j]=A[i];
                A[i]=aux;
            }
        }
    }
}

```

$S = (i, j, A)$

$(i, j, A) \rightarrow (i, j+1, A')$

$A[i] < A[j]$

Invariante

Estado inicial  $(1, N, A)$

Arreglo cualquiera

Estado final  $(N, N, A)$

Arreglo ordenado

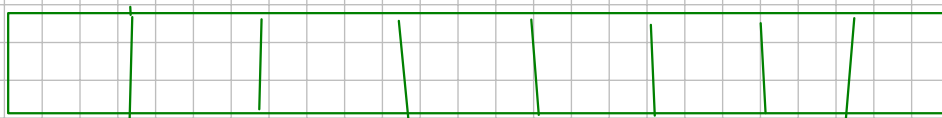
Invariante

1) Estado inicial, se tiene sólo un arreglo A arbitrario y depende de la entrada del usuario

2) Transformación de estado. Se sabe que el iterador sigue se va a cumplir  $A[i] \leq A[j]$  por el condicional.

3) Estado final, sé que la entrada debe estar ordenada.

¿Que cambia a medida que el algoritmo itera?



$A[i] \leq A[j]$

$S_k \rightarrow S_{k+1}$