

Ejemplos Invariantes de ciclo

Problema de Ordenamiento

Entrada: Un arreglo $A[0..N]$ de números, $N \geq 0$.

Salida: El arreglo A ordenado ascendente

```

1. def ordenar(A):
2.     i = 1
3.     while i < len(A):
4.         j = i - 1
5.         while j ≥ 0 and A[j+1] < A[j]:
6.             A[j], A[j+1] = A[j+1], A[j]
7.             j -= 1
8.         i += 1

```

Invariantes:

$I_0: 1 \leq i \leq N$

$I_1: \text{El arreglo } A[0..i) \text{ está ordenado}\text{ ascendente}$

Para demostrar que los invariantes I_0, I_1 se cumplen es necesario poder establecer que el ciclo while interna realiza su trabajo apropiadamente. Para esto se requiere hacer un análisis de instantes para ese ciclo:

```

4.     j = i - 1
5.     while j ≥ 0 and A[j+1] < A[j]:
6.         A[j], A[j+1] = A[j+1], A[j]
7.         j -= 1

```

Invariantes:

$I_2: -1 \leq j \leq i - 1$

$I_3: A[i] \text{ es el primer valor y el menor}\text{ valor del subarreglo } A[j+1..i+1]$

Teorema Invariantes I_2 y I_3 : Los invariantes I_2 y I_3 se cumplen.

Demarcación: Se procede mostrando la validez de los invariantes para la inicialización y la estabilidad.

Inicialización: De acuerdo a la línea 4 inicialmente $j = i-1$. Para los invariantes I_2 y I_3 se tiene que:

$$\begin{aligned} -1 &\leq j \leq i-1 \\ -1 &\leq i-1 \leq i-1 \end{aligned}$$

y el elemento $A[j+1] = A[i-1+1] = A[i]$ es el menor elemento del arreglo

$$A[j+1..i+1] = A[i-1+1..i+1] = A[i..i+1]$$

y por ende es trivialmente el menor valor.

Por lo tanto, los invariantes I_2 y I_3 se cumplen en la inicialización.

Estabilidad: Se considera una iteración arbitraria en la que $j = k$ y se asume que antes de ejecutar esta iteración los invariantes son ciertos. De esta manera, se tiene que:

$$\begin{aligned} -1 &\leq j \leq i-1 \\ -1 &\leq k \leq i-1 \end{aligned}$$

se cumple y que $A[i]$ es el primer y el menor valor en el subarreglo $A[j+1..i+1] = A[k+1..i+1]$. Además, dado que no es la última iteración se tiene que $j \geq 0$ y $A[j+1] < A[j]$, osea, $A[k+1] < A[k]$, osea $A[i] < A[k]$.

El objetivo es mostrar que después de esta iteración, osea, antes de la iteración donde $j = k-1$ los invariantes siguen siendo ciertos. Al ejecutar las líneas 6-7:

$$\begin{aligned} A[j], A[j+1] &\leftarrow A[j+1], A[j] & j = 1 \\ A[k], A[k+1] &\leftarrow A[k+1], A[k] & j = k-1 \end{aligned}$$

se intercambiarán los elementos en las posiciones k y $k+1$ en A . Dado que se tiene que $A[k+1] = A[i]$ es el menor elemento en $A[k+1 \dots i+1]$ y que $A[i] < A[k]$ entonces $A[i]$ es el menor elemento en $A[k \dots i+1]$.

Luego, el intercambio de la línea 6 ocasionará que este valor quede en la primera posición en $A[k \dots i+1]$.

Por lo tanto, el invariante I_3 seguirá siendo válido. Como $j \geq 0$, osea $k \geq 0$, después de ejecutar la línea 7 el invariante I_2 también seguirá siendo válido.

Finalmente, el ciclo terminará ya que el valor de j se reduce en 1 y eventualmente llegará a -1 o se llegará a un valor j tal que $A[j+1] > A[j]$ y por la estabilidad de los invariantes $A[i]$ será el menor elemento en $A[j+1 \dots i+1]$.

Teorema Invariantes I_0 y I_1 : Los invariantes I_0 y I_1 se cumplen.

Demarcación: Se procede mostrando la validez de los invariantes para la inicialización y la estabilidad.

Inicialización: De acuerdo a la línea 2 inicialmente $i=1$. Para los invariantes I_0 y I_1 se tiene que:

$$\begin{aligned} 1 \leq i \leq N \\ 1 \leq 1 \leq N \end{aligned}$$

y el arreglo $A[0..i] = A[0..1]$ contiene únicamente el elemento $A[0]$ y por lo tanto está trivialmente ordenado.

Por lo tanto, los invariantes I_0 y I_1 se cumplen en la inicialización.

Estabilidad: Se considera una iteración arbitraria en la que $i=k$ y se asume que antes de ejecutar esta iteración los invariantes son ciertos. De esta manera, se tiene que:

$$\begin{aligned} 1 \leq i \leq N \\ 1 \leq k \leq N \end{aligned}$$

es válido y el arreglo $A[0..k]$ está ordenado ascendente. Además, dado que no es la última iteración se tiene que $k < N$. El objetivo es mostrar que después de esta iteración, o sea, antes de la iteración donde $i = k+1$ los invariantes siguen siendo ciertos. Al ejecutar las líneas 4-8:

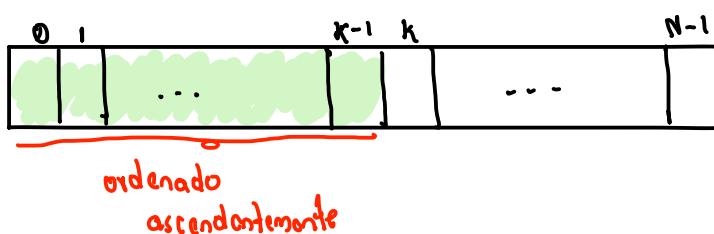
```

j = i-1
while j ≥ 0 and A[j+1] < A[j];
    A[j], A[j+1] = A[j+1], A[j]
    j -= 1
    i += 1

```

Lucgo, por la correctitud del ciclo while interno se tiene que al finalizar ese ciclo j será igual a -1 o será igual a algún h tal que $A[h]$ será el menor elemento en $A[h+1..N-1]$. Dado que $A[0..k]$ está ordenado ascendente todos los elementos en $A[0..h+1]$ serán menores que $A[h]$ y el arreglo $A[0..h+1]$ estará también ordenado ascendente y el invariante j , se sigue cumpliendo.

Antes de la iteración $i=k$



Después de la iteración cuando $i=k+1$



Al ejecutar la línea 8:

```

i += 1
i = k+1

```

como $k < N$ entonces $k+1 \leq N$ y en consecuencia el invariante I_0 sigue siendo válido.

Finalmente, el ciclo terminará ya que i empieza en 1 y su valor aumenta de uno en uno y eventualmente llegará a N . En ese punto, por la estabilidad del invariante I_0 , el arreglo $A[0..i] \subseteq A[0..N]$ estará ordenado ascendente.,,

Teorema: La invocación `ordenar(A)` para cualquier arreglo $A[0..N]$ ordena ascendente los elementos de A .

Demonstración: Es trivial a partir de la correctitud de los invariantes I_0 y I_1 .,,

Problema de División

Entrada: Número natural $a \geq 0$ y número natural $b > 0$.

Salida: (q, r) tal que $a = b \cdot q + r$ y $r < b$

```
1. pair<int,int> dividir(int a, int b){  
2.     int q=0, r=a;  
3.     while(r>b){  
4.         q+=1;  
5.         r-=b;  
6.     }  
7.     return {q,r};  
8. }
```

Invariantes:

$$I_0: a = b \cdot q + r$$

Teorema Invariante I_0 : El invariante I_0 se cumple.

Demonstración: Se procede mostrando la validez de los invariantes para la inicialización y la estabilidad.

Inicialización: De acuerdo a la línea 2 inicialmente $q=0$ y $r=a$. Para el invariante I_0 se tiene que:

$$\begin{aligned}a &= b \cdot q + r \\a &= b \cdot 0 + a \\a &= a\end{aligned}$$

Por lo tanto, el invariante I_0 se cumple en la inicialización.

Estabilidad: Se considera una iteración arbitraria en la que $q=q'$ y $r=r'$ y se asume que antes de ejecutar esta iteración los invariantes son ciertos. De esta manera, se tiene que:

$$\begin{aligned}a &= b \cdot q + r \\a &= b \cdot q' + r'\end{aligned}$$

Luego, como no es la última iteración entonces $r \geq b$, o sea, $r' \geq b$ y se tiene que:

$$r' = r' - b + b$$

El objetivo es mostrar que después de esta iteración, o sea, antes de la iteración donde $i=k+1$ los invariantes siguen siendo ciertos. Al ejecutar las líneas 4-5:

$$\begin{array}{ll}q := 1; & r := b; \\q = q + 1 & r = r - b \\q = q' + 1 & r = r' - b\end{array}$$

Posteriormente, al evaluar el invariante I_0 se tiene que:

$$\begin{aligned}a &= b \cdot q + r \\a &= b \cdot (q' + 1) + (r' - b) \\a &= b \cdot q' + b + r' - b \\a &= b \cdot q' + r'\end{aligned}$$

y esto es válido por el supuesto inicial de la estabilidad y por lo tanto el invariante I_0 sigue siendo válido.

Finalmente, el ciclo finalizará puesto que r inicia en a y en cada iteración se reduce en b por lo que eventualmente será menor que b . En este punto, por la estabilidad del invariante I_0 se tiene que:

$$a = q \cdot b + r$$

con $r < b$ como se requiere en la postcondición. //

Teorema: La invocación `divdir(a,b)` con cualquier $a \geq 0$ y $b > 0$ produce la pareja (q,r) correspondientes al cociente y residuo de dividir a entre b .

Demostación: Es trivial a partir de la correctitud de los invariantes I_0 y I_1 . //