Invariantes de ciclo

Demostrar lo que un ciclo mantiene

Carlos Andrés Delgado Saavedra

Pontificia Universidad Javeriana Cali

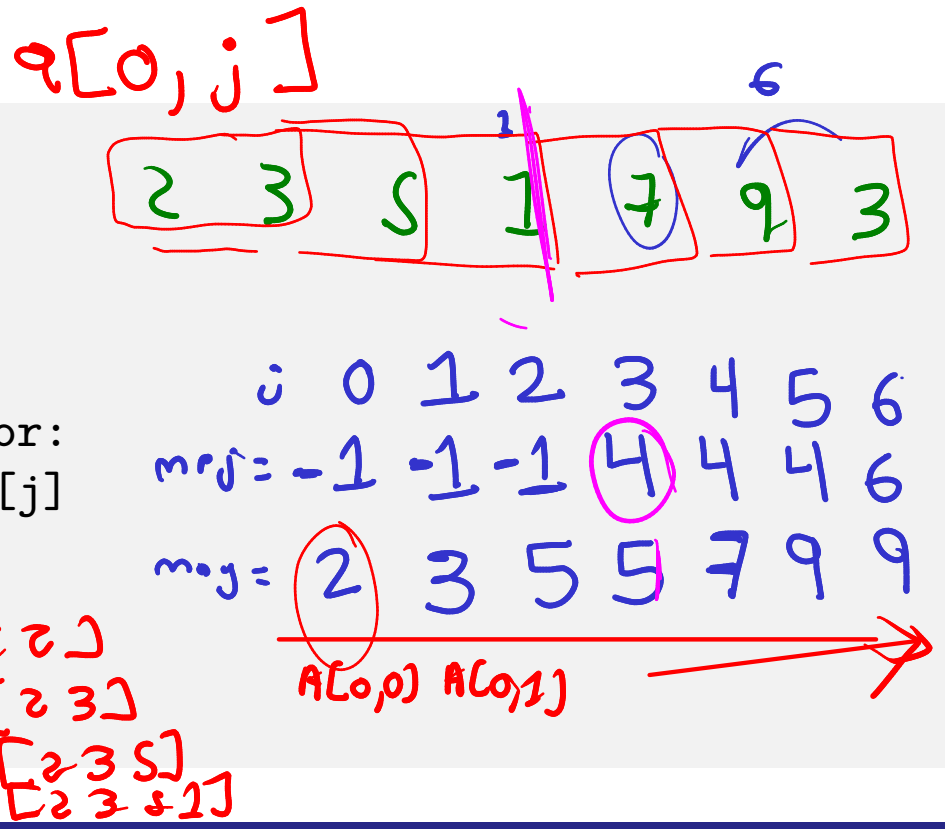
Contenido

1 Invariante de ciclo
- Concepto
- Ejemplo: el invariante de mayor
- Ejemplo: el invariante de mejor
- Ejercicio: UVa 507

2 Referencias

UVa 11078: ¿cómo se sabe que devuelve lo que debe?

```
0  def mayor_diferencia(a):
1      mayor = a[0]
2      mejor = a[0] - a[1]
3      j = 1
4      while j < len(a):
5          if mayor - a[j] > mejor:
6              mejor = mayor - a[j]
7          if a[j] > mayor:
8              mayor = a[j]
9          j = j + 1
10     return mejor
```



Especificación

Entrada: $a[0..n-1]$ de enteros, $n \geq 2$.

Salida: $\max\{\, a[s] - a[t] : 0 \leq s < t < n \,\}$.

$s \in [0, n-1) \wedge t \in [s+1, n), \; s \neq t$

Probar con ejemplos

¿Basta con probarlo en ejemplos? Este otro ciclo tiene los dos `if` al revés. Con $a = [1, 9, 2, 8, 3]$ devuelve 7, lo mismo que el original.

```
4 while j < len(a):
5     if a[j] > mayor:
6         mayor = a[j]
7     if mayor - a[j] > mejor:
8         mejor = mayor - a[j]
9     j = j + 1
```

Probar con ejemplos

j = 0

```
4 while j < len(a):
5     if a[j] > mayor:
6         mayor = a[j]
7     if mayor - a[j] > mejor:
8         mejor = mayor - a[j]
9     j = j + 1
```

a	original	al revés
$[1, 9, 2, 8, 3]$	7	7
$[1, 2, 3]$	-1	0

No basta: un ejemplo que pasa no dice nada de los demás arreglos. Hace falta un argumento que los cubra todos.

Invariante de ciclo

Definición

Un **invariante** es una fórmula sobre las variables del ciclo que es verdadera cada vez que se evalúa la condición del `while`. Se escribe con cuantificador, índices y rango.

Ejemplo, el índice del ciclo de UVa 11078, que arranca en 1, sube de 1 en 1 y sale al llegar a n:

$$I_0 : \quad 1 \leq j \leq n$$

Los tres pasos

Para demostrar que un invariante se cumple

1. **Inicialización**: es verdadero en el primer chequeo de la condición.
2. **Estabilidad**: si es verdadero en un chequeo y la condición se cumple, es verdadero en el siguiente.
3. **Terminación**: el ciclo termina, y el invariante junto con la condición falsa da lo que se quería.

Los dos primeros son una inducción sobre el número de chequeos: el caso base y el paso inductivo.

mayor: la traza

j	tramo $a[0..j)$	*mayor*

Con $a = [1, 9, 2, 8, 3]$, ¿qué vale *mayor* cada vez que la línea 4 evalúa la condición?

mayor: la traza

j	tramo $a[0..j)$	*mayor*
1	[1]	1

Primer chequeo. Las líneas 1 y 3 dejan $mayor = a[0] = 1$ y $j = 1$.

mayor: la traza

j	tramo $a[0..j)$	*mayor*
1	$[1]$	1
2	$[1, 9]$	9

Entra $a[1] = 9$, que supera a 1. La línea 8 cambia *mayor*.

mayor: la traza

j	tramo $a[0..j)$	*mayor*
1	$[1]$	1
2	$[1, 9]$	9
3	$[1, 9, 2]$	9

Entra $a[2] = 2$, que no supera a 9. *mayor* sigue igual.

mayor: la traza

j	tramo $a[0..j)$	*mayor*
1	$[1]$	1
2	$[1, 9]$	9
3	$[1, 9, 2]$	9
4	$[1, 9, 2, 8]$	9

Entra $a[3] = 8$, que tampoco supera a 9.

mayor: la traza

j	tramo $a[0..j)$	*mayor*
1	$[1]$	1
2	$[1, 9]$	9
3	$[1, 9, 2]$	9
4	$[1, 9, 2, 8]$	9
5	$[1, 9, 2, 8, 3]$	9

Último chequeo: $j = 5 = n$ y el ciclo termina. ¿Qué cumple *mayor* en todas las filas?

mayor: el invariante

En todas las filas, *mayor* es el máximo del tramo $a[0..j)$. Con cuantificadores y rango:

$$I_0: \quad 1 \le j \le n$$

$$I_1: \quad \exists t\,(0 \le t < j \wedge a[t] = mayor)$$
$$\wedge \ \forall t\,(0 \le t < j \to a[t] \le mayor)$$

$0 \le j < n$

mayor = max(A[0, j)

La primera parte de I_1 dice que *mayor* está en el tramo; la segunda, que nadie del tramo lo supera. Se abrevia $mayor = \text{máx}\, a[0..j)$.

Existe un valor entre A[0,j) tal que es igual a mayor
y la segunda que para todos los valores son menores o iguales a
a mayor

mayor: demostración

Teorema 1 y estrategia

Para todo $a[0..n-1]$ con $n \geq 2$, I_0 e I_1 valen en cada chequeo de la línea 4. Se demuestra con los tres pasos.

```
1 mayor = a[0]
2 mejor = a[0] - a[1]
3 j = 1
```

Inicialización. Las líneas 1 y 3 dejan $mayor = a[0]$ y $j = 1$.

- I_0: $1 \leq 1 \leq n$, porque $n \geq 2$. ✓
- I_1: el rango $0 \leq t < 1$ solo tiene $t = 0$, que es el testigo del $\exists$, y $a[0] \leq a[0]$. ✓

mayor: demostración

```
7 if a[j] > mayor:
8     mayor = a[j]
9 j = j + 1
```

Estabilidad: lo que se asume y lo que se pide

Una vuelta cualquiera, con $1 \leq j < n$ porque la condición se cumple. Se asumen I_0 e I_1 en j. I_1 en $j+1$ pide lo mismo sobre un tramo con una posición más, la $t = j$.

$j \longrightarrow j+1$

mayor: demostración

```
7 if a[j] > mayor:
8     mayor = a[j]
9 j = j + 1
```

Asumimos que para t = j existe un mayor

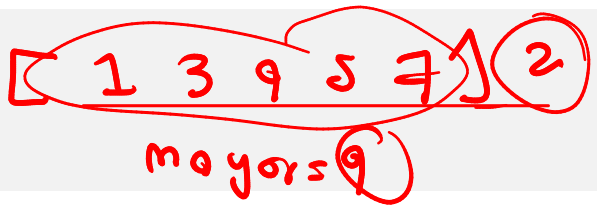
Estabilidad: caso $a[j] > mayor$

La línea 8 deja $mayor = a[j]$.

- Testigo del $\exists$: $t = j$. ✓
- Para $t < j$: $a[t] \leq$ el *mayor* viejo $< a[j]$, por I_1 en j. ✓
- Para $t = j$: $a[j] \leq a[j]$. ✓

mayor: demostración

```
7 if a[j] > mayor:
8     mayor = a[j]
9 j = j + 1
```



Estabilidad: caso $a[j] \leq mayor$

mayor no cambia.

- Testigo del $\exists$: el mismo de I_1 en j, que sigue en el rango. ✓
- Para $t < j$: por I_1 en j. Para $t = j$: es el caso. ✓

En los dos casos, la línea 9 lleva j a $j + 1$, y de $1 \leq j < n$ sale $1 \leq j + 1 \leq n$, que es I_0. ✓

mayor: demostración

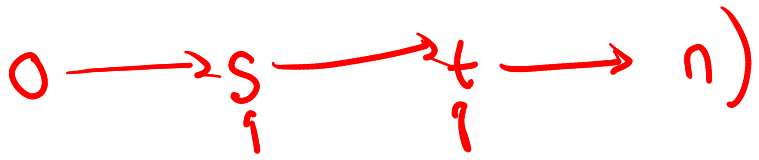
Terminación

j sube de 1 en 1 e I_0 lo acota por n, así que el ciclo termina. Al salir, $j < n$ es falsa, o sea $j \geq n$; con I_0 queda $j = n$. Sustituyendo en I_1:

$$mayor = \text{máx}\, a[0..n).$$

Por lo tanto, I_0 e I_1 valen en cada chequeo. ■

mejor: un candidato



mejor parece llevar la mayor diferencia entre las parejas del tramo $a[0..j)$:

$$I_2' : \quad \exists s\, \exists t\, (0 \leq s < t < j \wedge a[s] - a[t] = mejor)$$
$$\wedge\ \forall s\, \forall t\, (0 \leq s < t < j \rightarrow a[s] - a[t] \leq mejor)$$

¿Es un invariante? Compruébelo con $a = [1, 9, 2, 8, 3]$ en el primer chequeo.

mejor: un candidato

No: falla la inicialización

En el primer chequeo $j = 1$, y el rango $0 \leq s < t < 1$ no tiene parejas. El $\exists$ sobre un rango vacío es falso, sin importar lo que valga *mejor*. Pero la línea 2 ya guardó $a[0] - a[1] = -8$, la diferencia de la pareja $(0, 1)$, que todavía no está en el tramo.

Desde $j = 2$ el candidato sí se cumple. Lo que le falta es contar esa pareja desde el principio.

mejor: el invariante

Las parejas que *mejor* ya tuvo en cuenta son la de la línea 2 y las del tramo:

$$P_j = \{(0,1)\} \cup \{\, (s,t) : 0 \le s < t < j \,\}$$

$$I_2 : \quad \exists (s,t) \in P_j \; (a[s] - a[t] = \mathit{mejor})$$
$$\wedge \; \forall (s,t) \in P_j \; (a[s] - a[t] \le \mathit{mejor})$$

Se abrevia $\mathit{mejor} = \text{máx}\{\, a[s] - a[t] : (s,t) \in P_j \,\}$.

mejor: el invariante

j	*mayor*	*mejor*	la pareja que lo alcanza
1	1	-8	$(0,1)$: $1-9$
2	9	-8	$(0,1)$: $1-9$
3	9	7	$(1,2)$: $9-2$
4	9	7	$(1,2)$: $9-2$
5	9	7	$(1,2)$: $9-2$

Con $a = [1, 9, 2, 8, 3]$. El 7 sale de restar $a[2]$ al *mayor* que había en $j = 2$.

mejor: la mejor pareja nueva

Cada vuelta, con $1 \leq j < n$, agrega las parejas (s, j) con $0 \leq s < j$.

La mejor pareja nueva sale de I_1

La parte $\exists$ de I_1 da un $t_0 < j$ con $a[t_0] = \mathit{mayor}$. La parte $\forall$ da, para todo $s < j$,

$$a[s] - a[j] \;\leq\; \mathit{mayor} - a[j] \;=\; a[t_0] - a[j].$$

La mejor pareja nueva es (t_0, j), con diferencia $\mathit{mayor} - a[j]$.

Por eso *mejor* depende de *mayor*, que las líneas 5 y 6 leen antes de que la 8 lo cambie. La demostración completa de I_2 va con la solución del ejercicio.

1. Hacer una prueba de escritorio

2. Identificar que es lo que cambia en cada iteración, siempre hay un contador (este o no) para contar la iteración de alli

I0

3. Cuando se identifique el cambio, la invariante describe contenido y condiciones
a) El valor que contiene es el mayor del segmento evaluado
b) El valor que contiene es una acumulación (sumatoria)
c) El valor es un valor que depende de una condición
d) El valor que contiene es relacionado con otros valores (diferencia)

Descripción: valor que contiene
Condición: Como es el valor frente a los demas

Ejercicio en papel: UVa 507

```
0  def mejor_tramo(v):
1      suma = 0
2      mejor = 0
3      k = 0
4      while k < len(v):
5          suma = suma + v[k]
6          if suma > mejor:
7              mejor = suma
8          if suma < 0:
9              suma = 0
10         k = k + 1
11     return mejor
```

Especificación

Entrada: $v[0..n-1]$ de enteros, las calles de la ruta.
Salida: la mayor suma de un tramo de calles seguidas, 0 si ninguno es positivo.

Ejercicio en papel: UVa 507

Enunciado

Sea $S(i,j) = \sum_{t=i}^{j-1} v[t]$, con $S(i,i) = 0$, e I_0: $0 \leq k \leq n$.

1. Llene la traza que sigue.
2. Escriba I_1, sobre *suma*, con $\exists$, $\forall$ y rango.
3. Escriba I_2, sobre *mejor*, de la misma forma.
4. Demuestre inicialización, estabilidad y terminación.

Ejercicio en papel: UVa 507

k	$v[k]$	*suma*	*mejor*
0	2	0	0
1	-3		
2	4		
3	-1		
4	2		
5	-6		
6			

Cada fila es un chequeo de la línea 4, antes de ejecutar la vuelta.

Ejercicio en papel: UVa 507

Consejo

- Donde las dos columnas difieren, *suma* no es lo mejor de lo recorrido: mire dónde terminan sus tramos.
- Pruebe cada fórmula en el primer chequeo, $k = 0$: ahí cayó el candidato de *mejor*.
- En la estabilidad cada `if` va por casos, y uno usa el invariante del otro.

Contenido

1 Invariante de ciclo

2 Referencias

Referencias

- K. H. Rosen. *Matemáticas Discretas y sus Aplicaciones*, 7.ª edición, McGraw-Hill. Sección 5.5: correctitud de programas e invariantes de ciclo.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest y C. Stein. *Introduction to Algorithms*, 3.ª edición, MIT Press, 2009. Sección 2.1, pp. 18–20: invariantes de ciclo, donde la estabilidad se llama *mantenimiento*.

Problemas

UVa 11078, *Open Credit System*:
`https://vjudge.net/problem/UVA-11078`
UVa 507, *Jill Rides Again*:
`https://vjudge.net/problem/UVA-507`
Envío del 507:
`https://vjudge.net/problem/UVA-507`