



aprenderaprogramar.com

Ejercicio ejemplo de verificación de algoritmos. Análisis de eficiencia en programación. (CU00239A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 38 del Curso Bases de la programación Nivel II

24

EJERCICIO

Nos hemos planteado el siguiente objetivo: “Queremos leer datos desde un archivo que contiene un array $Dato(1), Dato(2), \dots, Dato(n)$ en paquetes de 5 datos. Contaremos el número de datos cuyo valor sea inferior a 100 (a los que llamaremos datos válidos) de modo que si tras extraer un paquete el número de datos válidos es igual o superior a 5 el algoritmo finaliza”.

Determinar cuál de los dos algoritmos que se plantean cumple el objetivo con eficiencia.

Algoritmo 1:

```
1.  $i = 0 : A = 1 : B = 5$ 
2. Mientras  $i \leq 5$  Hacer
    3. Desde  $j = A$  hasta  $B$  Hacer
        4. Leer  $Dato(j)$ 
        5. Si  $Dato(j) < 100$  Entonces
            6.  $i = i + 1$ 
        7. FinSi
    8. Siguiente  $j$ 
    9.  $A = B : B = B + 5$ 
10. Repetir
```

Algoritmo 2:

```
1.  $i = 0 : A = 1 : B = 5$ 
2. Mientras  $i \leq 5$  Hacer
    3. Desde  $j = A$  hasta  $B - 1$  Hacer
        4. Leer  $Dato(j)$ 
        5. Si  $Dato(j) < 100$  Entonces
            6.  $i = i + 1$ 
        7. FinSi
    8. Siguiente  $j$ 
    9.  $A = B : B = B + 5$ 
10. Repetir
```

SOLUCIÓN

El ejercicio tiene un poco de enjundia, ya que el enunciado contiene una pequeña trampa. Pero vayamos al asunto. Supuestamente uno de los algoritmos es incorrecto. En teoría deben leer paquetes de 5 datos hasta que el número de datos válidos es igual o superior a 5. Se nos ocurren dos grupos de casos:

- Los cinco primeros datos son válidos.
- Los cinco primeros datos no son válidos.

En el caso a) sólo se leería un paquete de datos (5 datos), ya que una vez realizado el proceso se cumple la condición de salida. En el caso b) habrá que extraer, al menos, dos paquetes de datos (10 datos). En primer lugar comprobaremos si los dos algoritmos resuelven satisfactoriamente el caso a), o sólo uno de ellos.

<i>Algoritmo 1. Dato(1) < 100, Dato(2) < 100, ..., Dato(n) < 100</i>					
Estado	i	A	B	j	Número de lecturas
Inicio Línea 1	0	1	5	0	~
Evaluación bucle exterior Línea 2	0	~	~	~	~ [Entra bucle exterior]
Líneas 3 – 4 – 5 – 6 – 7	1	~	~	1	1 [Entra bucle j]
Líneas 8 – 3 – 4 – 5 – 6 – 7	2	~	~	2	2
Líneas 8 – 3 – 4 – 5 – 6 – 7	3	~	~	3	3
Líneas 8 – 3 – 4 – 5 – 6 – 7	4	~	~	4	4
Líneas 8 – 3 – 4 – 5 – 6 – 7	5	~	~	5	5
Línea 8 – 3	~	~	5	6	~ [Sale bucle j]
Línea 9 – 10 – 2	5	5	10	~	~
Líneas 3 – 4 – 5 – 6 – 7	6	~	~	5	6 [Entra bucle j] [Repite lectura de dato 5]
Líneas 8 – 3 – 4 – 5 – 6 – 7	7	~	~	6	7 [Lectura de dato 6]
ALGORITMO SIGUE PERO NO CUMPLE OBJETIVOS					

<i>Algoritmo 2. Dato(1) < 100, Dato(2) < 100, ..., Dato(n) < 100</i>						
Estado	i	A	B	j	B – 1	Número de lecturas
Inicio Línea 1	0	1	5	0	4	~
Evaluación bucle exterior Línea 2	0	~	~	~	~	~ [Entra bucle exterior]
Líneas 3 – 4 – 5 – 6 – 7	1	~	~	1	~	1 [Entra bucle j]
Líneas 8 – 3 – 4 – 5 – 6 – 7	2	~	~	2	~	2
Líneas 8 – 3 – 4 – 5 – 6 – 7	3	~	~	3	~	3
Líneas 8 – 3 – 4 – 5 – 6 – 7	4	~	~	4	4	4
Línea 8 – 3	~	~	~	5	4	~ [Sale bucle j]

Línea 9 – 10 – 2	4	5	10	~	9	~
Líneas 3 – 4 – 5 – 6 – 7	5	~	~	5	~	5
Líneas 8 – 3 – 4 – 5 – 6 – 7	6	~	~	6	~	6
ALGORITMO SIGUE PERO NO CUMPLE OBJETIVOS						

Aquí estaba la “envidia” y la trampa del enunciado: ninguno de los dos algoritmos cumple el objetivo. El algoritmo 1 lee los 5 datos con lo cual parece que maneja bien los “paquetes” de datos. Sin embargo, aunque los cinco datos sean válidos no se para la extracción, incumpliendo el objetivo. El algoritmo 2 es similar pero en la línea 2 el bucle *Desde* tiene límite superior $B - 1$ en vez de B . Parece querer decir “como el otro algoritmo se pasa, rebajo el límite del bucle para no pasarme”. Pensamiento un tanto errático que lleva a que en vez de un paquete de 5 datos se extraigan 4 datos entre otras cosas..., incumpliendo también el objetivo. Vamos a ver un algoritmo que funciona correctamente:

Algoritmo 3:

```

1.  $i = 0 : A = 1 : B = 6$ 
2. Mientras  $i < 5$  Hacer
    3. Desde  $j = A$  hasta  $B - 1$  Hacer
        4. Leer Dato(j)
        5. Si Dato(j) < 100 Entonces
            6.  $i = i + 1$ 
        7. FinSi
    8. Siguiente j
    9.  $A = B : B = B + 5$ 
10. Repetir
    
```

Los cambios introducidos han sido:

- Se define $B = 6$ en vez de $B = 5$.
- La condición $i \leq 5$ cambia a $i < 5$.
- El límite superior del bucle j pasa de ser B a $B - 1$.

Comprobemos que para un paquete inicial de 5 datos válidos el bucle funciona correctamente:

<i>Algoritmo 3. Dato(1) < 100, Dato(2) < 100, ..., Dato(n) < 100</i>						
Estado	i	A	B	j	B - 1	Número de lecturas
Inicio Línea 1	0	1	6	0	5	~
Evaluación bucle exterior Línea 2	0	~	~	~	~	~ [Entra bucle exterior]
Líneas 3 – 4 – 5 – 6 – 7	1	~	~	1	5	1 [Entra bucle j]

Líneas 8 – 3 – 4 – 5 – 6 – 7	2	~	~	2	~	2
Líneas 8 – 3 – 4 – 5 – 6 – 7	3	~	~	3	~	3
Líneas 8 – 3 – 4 – 5 – 6 – 7	4	~	~	4	~	4
Líneas 8 – 3 – 4 – 5 – 6 – 7	5	~	~	5	5	5
Línea 8 – 3	~	~	~	6	5	~ [Sale bucle j]
Línea 9 – 10 – 2	5	6	11	~	~	~ [Sale bucle exterior]

En esta ocasión el funcionamiento es correcto, pues al leerse 5 datos válidos se sale del bucle. Téngase en cuenta que si se hubiera definido $B = 5$ y el bucle *Desde* de la línea 3 con límite superior B las lecturas serían (suponiendo datos no válidos):

1, 2, 3, 4, 5 (5 datos)
 5, 6, 7, 8, 9, 10 (6 datos)
 10, 11, 12, 13, 14, 15 (6 datos)
 Etc.

¿Por qué 6 datos? Porque en el segundo bucle repetimos como punto de partida el punto de llegada del anterior, cosa que evitamos en el algoritmo 3.

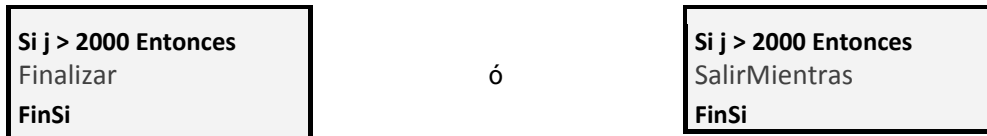
Pequeños detalles pueden trastocarnos el funcionamiento de grandes programas. A lo largo del curso hemos trabajado con algoritmos similares a éste y eso no significa que podamos resolver este tipo de algoritmos rápidamente. A pesar de su aparente sencillez requieren hilar fino. En este caso estaríamos, como ya nos ha pasado en más de una ocasión (ver ejercicio relativo a algoritmo para el manejo de una lista de datos), ante algo que intuimos pero cuyo método paso a paso desconocemos. Como ya expusimos, entra en juego la experiencia del programador y el uso de las técnicas expuestas en “Conocer el problema”.

Estudiando el funcionamiento del algoritmo 3 llegamos a la siguiente tabla:

Ratio de datos válidos	Datos leídos	Datos válidos leídos	Pasadas por el bucle exterior
Constante 5 / 5	5	5	1
Constante 4 / 5	10	8	2
Constante 3 / 5	10	6	2
Constante 2 / 5	15	6	3
Constante 1 / 5	25	5	5

Constante 0 / 5	∞	0	∞
Indefinido	Indefinido	Indefinido	Indefinido

En esta tabla nos ha de llamar la atención que para una posibilidad de datos de entrada aparezca “pasadas por el bucle exterior = ∞ ”. El famoso bucle infinito. Convendría preverlo y evitar que se pueda producir. Para ello bastaría introducir en el bucle una cláusula de seguridad tipo:



Este u otro mecanismo deben protegernos también del número de repeticiones “indefinido”. Si la cantidad de datos válidos es muy baja en relación al total, la salida del bucle se puede demorar. Podría darse un mensaje de advertencia al usuario para advertirle de que sólo se han encontrado x datos válidos tras tantas miles de lecturas y pedir si se quiere continuar o desistir.

Como consideración final, a lo largo de los ejercicios anteriores hemos visto distintos enfoques de la verificación de algoritmos pero siempre usando tablas de variables por resultar estas bastante explicativas y claras. Sin embargo un programador no usa una sola técnica de verificación: las ha de conocer todas y recurrir a la más oportuna para una circunstancia dada. Muchas de las verificaciones que hemos visto podrían haberse resuelto con una verificación mental o por seguimiento escrito. De hecho, la mayoría de bucles y procesos simples los acabaremos resolviendo así por ser lo más rápido.

En algún caso, no hemos realizado un comprobación por requerir un número de iteraciones muy elevado, lo que un ordenador o dispositivo de programación nos puede resolver fácilmente.

Próxima entrega: CU00240A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60