Pregunta Bittercios Algoritmos Iterativos

Fecha de Creación	@13 de octubre de 2023 15:43
Asignatura	FAL
Fecha de Modificación	@16 de octubre de 2023 15:16

1. Recorrer el vector de INICIO a FIN comparando cada elemento salvo el PRIMERO con el elemento ANTERIOR
   a. P7: ordenado(a, 0, i-1, resul) ^ 0 < i ≤ n (ESTE FUE DONDE ME EQUIVOQUE)
2. Recorrer el vector de INICIO a FIN comparando cada elemento salvo el ULTIMO con el elemento SIGUIENTE
   a. P3: ordenado(a, 0, i, resul) ^ 0 ≤ i < n
3. Recorrer el vector de FIN a INICIO comparando cada elemento salvo el PRIMERO con el elemento ANTERIOR
   a. P1: ordenado(a,i,n-1,resul) ^ 0 ≤ i < n
4. Recorrer el vector de FIN a INICIO comparando cada elemento salvo el ULTIMO con el elemento SIGUIENTE
   a. P5: ordenado(a,i+1,n-1,resul) ^ -1 ≤ i < n - 1

Pregunta 1
Parcialmente correcta
Se puntúa 0,75 sobre 1,00
Marcar pregunta

Para comprobar si un vector de enteros está o no ordenado (en orden creciente), se plantean las siguientes estrategias:

- Estrategia 1: Recorrer el vector del principio hasta el final, comparando cada elemento (a excepción del primero) con el elemento anterior.
- Estrategia 2: Recorrer el vector del principio hasta el final, comparando cada elemento (a excepción del último) con el elemento siguiente.
- Estrategia 3: Recorrer el vector desde el final hasta el principio, comparando cada elemento (a excepción del primero) con el elemento anterior.
- Estrategia 4: Recorrer el vector desde el final hasta el principio, comparando cada elemento (a excepción del último) con el elemento siguiente.

Para abordar el diseño de algoritmos para cada una de las estrategias, por simplicidad, se asume que el vector debe contener al menos un elemento, que se introduce una variable de control **i** utilizada para iterar sobre el vector, y que se define un predicado *ordenado(a,i,j,resul)* como *resul* = ~~∀i,j:0≤i≤j<n:a[i]≤a[j]~~. (∀u,v: i≤u≤v≤j: a[u]≤a[v]) Debes elegir, de entre los siguientes predicados, un invariante adecuado para justificar la corrección de cada estrategia:

- P1: ordenado(*a,i,n-1,resul*) ∧ 0≤i<n
- P2: ordenado(*a,0,i,resul*) ∧ 0<i≤n
- P3: ordenado(*a,0,i,resul*) ∧ 0≤i<n
- P4: ordenado(*a,i+1,n-1,resul*) ∧ 0≤i<n
- P5: ordenado(*a,i+1,n-1,resul*) ∧ -1≤i<n-1
- P6: ordenado(*a,0,i,resul*) ∧ 0<i<n
- P7: ordenado(*a,0,i-1,resul*) ∧ 0<i≤n
- P8: ordenado(*a,i+1,n-1,resul*) ∧ 0≤i<n-1

Importante: no basta conque el predicado sea un invariante del proceso iterativo asociado a la estrategia, sino que, además, debe permitir justificar la corrección parcial de la misma.

¿Invariante para la estrategia 1? P4 ✗
¿Invariante para la estrategia 2? P3 ✓
¿Invariante para la estrategia 3? P5 ✓
¿Invariante para la estrategia 4? P1 ✓

Respuesta parcialmente correcta.

Ha seleccionado correctamente 3.
Estrategia 1: al comienzo de la iteración **i**-esima habremos determinado que el tramo a[0..i) (es decir, la secuencia de valores a[0] ... a[i-1]) está ordenado. Entonces, si a[i-1]≤a[i], continuaremos iterando, y, si no, decidiremos que el vector no está ordenado. Por tanto, **i** dejará siempre a su izquierda el segmento ya tratado (pero a[i] no formará parte de dicho segmento). Con ello, antes y después de cada iteración se cumplirá ordenado(*a,0,i-1,resul*) (ten en cuenta que este predicado indica que el tramo a[i..j], es decir a[i]...a[j], incluyendo los valores en los extremos, está ordenado). Se comenzará por la posición 1, ya que siempre comparamos con el anterior, y se saldrá del bucle al llegar a n. Por tanto, el invariante adecuado para esta estrategia es

P7: ordenado(*a,0,i-1,resul*) ∧ 0<i≤n

Estrategia 2: en este caso, dado que comparamos con el elemento siguiente, al comienzo de cada iteración podremos asegurar que el tramo a[0..i] (a[i] incluido) está ya ordenado. Entonces, si a[i]≤a[i+1], continuaremos iterando, y, si no, decidiremos que el vector no está ordenado. Comenzaremos en la posición 0, ya que comparamos con el siguiente, y abandonaremos el bucle cuando i sea n-1 (ya a[n-1] no tendrá siguiente). Por tanto, el invariante adecuado para esta estrategia es

P3: ordenado(*a,0,i,resul*) ∧ 0≤i<n

Estrategia 3: en este caso, dado que comparamos cada elemento con el anterior, al comienzo de cada iteración podremos asegurar que el tramo a[i,n) (a[i] incluido) está ya ordenado. Entonces, si a[i-1]≤a[i], continuaremos iterando, y, en caso contrario, decidiremos que el vector no está ordenado. Tenemos que comenzar en la posición n-1, y, cuando salgamos del bucle, i será 0. Por tanto, el invariante adecuado para esta estrategia será:

P1: ordenado(*a,i,n-1,resul*) ∧ 0≤i<n

Estrategia 4: en este caso, dado que comparamos cada elemento con el siguiente, al comienzo de cada iteración podremos asegurar que el tramo a(i,n) (a[i] excluido) está ya ordenado. Entonces, si a[i]≤a[i+1], continuaremos iterando, y, en caso contrario, decidiremos que el vector no está ordenado. Tenemos que comenzar en la posición n-2 (a[n-1] no tiene siguiente), y, cuando salgamos del bucle, i será -1 (porque, en general, hemos tenido que comparar a[0] con a[1]). Por tanto, el invariante adecuado para esta estrategia será:

P5: ordenado(*a,i+1,n-1,resul*) ∧ -1≤i<n-1.

La respuesta correcta es:
¿Invariante para la estrategia 1? → P7,

¿Invariante para la estrategia 2? → P3,

¿Invariante para la estrategia 3? → P5,

¿Invariante para la estrategia 4? → P1