Tarea 2 DALGO

Santiago Rangel

Andrés Felipe Losada

1.

Considere el problema de calcular la raíz cúbica entera de un número natural n de acuerdo con la

siguiente especificación:

[ Ctx C: n: nat

{Q: true}

...

do ... od

]

1a Escriba código GCL que siga la especificación indicada. Dentro del código use únicamente operaciones de suma, resta y multiplicación.

[ Ctx C: n: nat

{Q: true}

Do

;

;

od

1b Indique qué técnica(s) puede(n) explicar la definición del invariante P a partir de la especificación

dada.

Primero se usa la técnica de eliminar conjunción. Se elimina en R1 al proponerla como invariante de tal manera que solo se tenga que revisar la segunda condición. Segundo, se usa el fortalecimiento del invariante al poner para evitar el cálculo de potencias (pero no se usa para esta solución)

1c Cuente asignaciones como operaciones básicas y estime las complejidades espacial y temporal de su solución, justificando sus respuestas.

La complejidad especial de este algoritmo es de ( ya que no usa ninguna estructura de datos adicional. Solo se usan 3 variables. La complejidad temporal es de (. Si usamos las asignaciones como operaciones básicas solo se harán ( ya que se va subiendo b de a 1 hasta llegar a ese valor.

2.

a)

;

;

;

;

;

b)

* *Explique qué técnica (de desarrollo de invariantes) puede justificar el invariante correspondiente*

Las técnicas utilizadas en los invariantes son: fortalecer el invariante y reemplazar una constante por una variable. El primero es porque se le agrega información al invariante para facilitar el problema y acercarlo a la pos-condición. El segundo se usa creando varias variables, y entre ellas una que se dedica a modificar el arreglo de factoriales.

* *Explique por qué hay terminación.*

En ambos ciclos hay terminación porque la cota decrece en cada iteración. En el primer ciclo en cada iteración se incrementa por uno el f, y esto hace que la cota disminuya hasta que sea igual a n. En el segundo ciclo c siempre crece al ser usado como un contador que facilita recorrer el arreglo, al final c es igual que n y la acaba.

c)

* *Estime con expresiones θ(...) las complejidades temporal y espacial de su algoritmo. Justifique sus respuestas.*

La complejidad temporal es θ(n), ya que recorre dos veces hasta n y θ(2n) = θ(n).  
La complejidad espacial es igual, ya que se crean un arreglos de tamaño n, y pocas variables.

3. Demuestre que son semianillos:

a.

Es monoide conmutativo:

1. :
2. :
3. :

Es monoide:

1. : : (0/1 min 1 =1)
2. :
3. :

\* distribuye sobre + (a la izquierda y la derecha):

1. :
2. : :

El módulo es el 0 de la multiplicación:

1. :
2. :

Como cumple con las 4 reglas, entonces es un semianillo.

b.

Es monoide conmutativo:

1. :
2. : Si a>b>c
3. :

Es monoide:

1. :
2. :
3. : Si a>b>c

\* distribuye sobre + (a la izquierda y la derecha):

1. :
2. : :

El módulo es el 0 de la multiplicación:

1. :
2. :

Como cumple con las 4 reglas, entonces es un semianillo.

4. Complete los demás pasos para construir su solución (recurrencia, diagrama de necesidades, estructura de datos, invariante). No es necesario que escriba su algoritmo, pero sí que estime las complejidades espacial y temporal de su solución.

Lenguaje:

Recurrencia:

Diagrama de necesidades:

