

Comenzado el Monday, 20 de July de 2020, 16:58

Estado Finalizado

Finalizado en Monday, 20 de July de 2020, 20:01

Tiempo empleado 3 horas 2 minutos

Calificación Sin calificar aún

Pregunta 1

Finalizado

Puntúa como 25

Especificar el problema de `parMasGrande`, que dada una secuencia devuelve la posición del elemento par mayor en módulo. Por ejemplo, dada `<11,8,7,-8,2,3,-4>` se podría devolver 3.

<

.....

Pregunta 2

Correcta

Puntúa 5 sobre 5

Dado un problema especificado con la siguiente postcondición, indicar qué términos deberían estar necesariamente incluidos en la precondition (se busca la precondition más débil posible)

```
proc indiceMasGrande (in s: seq(Z), out res: Z) {  
  Pre {...}  
  Post {0 ≤ res < |s| ∧ (∀ i : Z) (0 ≤ i < |v| → v[i] ≤ v[res])}  
}
```

Seleccione una o más de una:

- ☐ a.
No es necesario agregar ningún término, la precondition podría ser True
- ☐ b. $(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow s[j] \geq 0)$
- ☐ c.
 $(\forall x : \mathbb{Z}) \#apariciones(s, x) \leq 1$
- ☒ d.
 $|s| > 0$



Respuesta correcta

La respuesta correcta es:

$|s| > 0$

Pregunta 3

Correcta

Puntúa 5 sobre 5

Dadas las siguientes especificaciones y un programa P1 que es correcto respecto a `especificacion1`:

```
proc especificacion1 (in v: seq(Z), out res: Z) {  
  Pre { |v| > 0 }  
  Post { res ∈ v ∧ (∀x: Z)(x ∈ v → x ≤ res) }  
}  
  
proc especificacion2 (in v: seq(Z), out res: Z) {  
  Pre { |v| > 1 }  
  Post { res ∈ v ∧ ¬(∃x: Z)(x ∈ v ∧ x > res) }  
}
```

Seleccione una o más de una:

- ☒ a.
Se puede afirmar que el programa P1 es correcto respecto a `especificacion2` porque las postcondiciones son equivalentes y la precondición de `especificacion1` es más débil que la precondición de `especificacion2`.
✓
- ☐ b.
No puede afirmarse con certeza si P1 es correcto respecto a `especificacion2`
- ☐ c.
Se puede afirmar que el programa P1 es correcto respecto a `especificacion2` porque la precondición de `especificacion2` es más débil que la precondición de `especificacion1`, y la postcondición de `especificacion2` es más fuerte que la postcondición de `especificacion1`.
- ☐ d.
Se puede demostrar que el programa P1 NO es correcto respecto a `especificacion2`

Respuesta correcta

La respuesta correcta es:

Se puede afirmar que el programa P1 es correcto respecto a `especificacion2` porque las postcondiciones son equivalentes y la precondición de `especificacion1` es más débil que la precondición de `especificacion2`.

Pregunta **4**

Sin contestar

Puntúa como 25

Sea el siguiente ciclo con su correspondiente precondition y postcondición:

$$P_c : \{i = 1 \wedge s = s_0\}$$

```
while (i < |s|) do
  if (s[i-1] < s[i])
    s[i-1] := s[i];
  else
    skip;
  endif;
  i:=i+1;
endwhile
```

$$Q_c : \{|s| = |s_0| \wedge (\forall i: \mathbb{Z})(1 \leq i < |s| \wedge s_0[i-1] < s_0[i] \rightarrow s[i-1] = s_0[i]) \wedge (\forall i: \mathbb{Z})(1 \leq i < |s| \wedge s_0[i-1] \geq s_0[i] \rightarrow s[i-1] = s_0[i-1])\}$$

Y sea el siguiente invariante:

$$I: \{1 \leq i \leq |s| \wedge |s| = |s_0| \wedge (\forall j: \mathbb{Z})(0 < j < i-1 \wedge s_0[j-1] < s_0[j] \rightarrow s[j-1] = s_0[j]) \wedge (\forall j: \mathbb{Z})(0 < j < i-1 \wedge s_0[j-1] \geq s_0[j] \rightarrow s[j-1] = s_0[j-1]) \wedge (\forall j: \mathbb{Z})(i-1 \leq j < |s| \rightarrow s[j] = s_0[j])\}$$

Con el invariante propuesto, demostrar que

a) (20 puntos) $\{I \wedge B\}$ ciclo $\{I\}$

b) (5 puntos) $\{I \wedge \neg B\} \rightarrow Q_c$

.....

Pregunta 5

Incorrecta

Puntúa 0 sobre 5

Dado el siguiente programa en SmallLang, precondition y postcondition del ciclo:


```
i := 0;
while (i < v.size()) do
  v[i] := v[i] + 1;
  i = i + 1;
endwhile
```

- $P_e : i = 0 \wedge v = V_0$
- $Q_e : 0 \leq i \leq |v| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |v| \rightarrow_L v[j] = V_0[j] + 1)$

Es correcto el siguiente invariante para el programa dado?

$I : 0 \leq i \leq |v| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L v[j] = V_0[j] + 1)$

Seleccione una o más de una:

- ☐ a.
No, debería agregarse el siguiente término para poder demostrar que el cuerpo del ciclo preserva el invariante:
 $(\forall j : \mathbb{Z})(i \leq j < |v| \rightarrow_L v[j] = V_0[j])$
- ☒ b.
Sí, es correcto y permite demostrar el teorema del invariante.

- ☐ c.
No, debería modificarse el cuantificador de la siguiente forma para poder demostrar que el cuerpo del ciclo preserva el invariante:
 $(\forall j : \mathbb{Z})(0 \leq j \leq i \rightarrow_L v[j] = V_0[j] + 1)$
- ☐ d.
No, debería modificarse el rango de i de la siguiente forma para poder demostrar que el cuerpo del ciclo preserva el invariante: $0 \leq i < |v|$

Respuesta incorrecta.

La respuesta correcta es:

No, debería agregarse el siguiente término para poder demostrar que el cuerpo del ciclo preserva el invariante:

$(\forall j : \mathbb{Z})(i \leq j < |v| \rightarrow_L v[j] = V_0[j])$

Pregunta 6

Correcta


Puntúa 5 sobre 5

Dada la siguiente instrucción **S**

S: $s[i] := s[j] + s[i-1]$

Decidir si $\text{def}(S) \equiv 0 \leq i < |s| \wedge 0 \leq j < |s|$

Seleccione una:

- ☐ Verdadero
- ☒ Falso 

La respuesta correcta es 'Falso'



Pregunta **7**

Incorrecta

Puntúa 0 sobre 5

Indique cual/es de las siguientes afirmaciones es verdadera:

Seleccione una o más de una:

- ☒ a. Es posible armar un test suite que cubra todas las condiciones básicas de un programa pero que no cubra todos los branches del mismo. 
- ☐ b. Si un test suite t1 tiene más casos de test que un test suite t2, entonces t1 cubrirá más líneas (o igual cantidad) que t2.
- ☒ c. Es posible armar un test suite que cubra todas las líneas de un programa pero que no detecte errores en el mismo. 

Respuesta incorrecta.

La respuesta correcta es: Es posible armar un test suite que cubra todas las líneas de un programa pero que no detecte errores en el mismo.

Pregunta **8**

Correcta

Puntúa 5 sobre 5

El siguiente programa pretende poner el valor 1 en aquellas posiciones donde hubiera números positivos, 2 donde hubiera cero, y 3 donde hubiera números negativos.

```
void UnoDosTres(vector<int> v){  
L1: int i = 0;  
L2: while(i< v.size()){  
L3:   if(v[i] > 0){  
L4:     if(v[i] != 0){  
L5:       v[i] = 1;  
        }else{  
L6:       v[i] = 2;  
        }  
        }else{  
L7:       v[i] = 3;  
        }  
L8:   i++;  
        }  
}
```

Señale qué líneas del programa no se pueden cubrir con ningún test.

Seleccione una o más de una:

- ☐ L1
- ☐ L2
- ☐ L3
- ☐ L4
- ☐ L5
- ☒ L6 
- ☐ L7
- ☐ L8

Respuesta correcta

La respuesta correcta es: L6

Pregunta **9**

Correcta

Puntúa 5 sobre 5

Se tiene un programa P1 que recibe como parámetro un vector de tamaño n . Sabemos que el programa tiene dos ciclos anidados. ¿Qué podemos decir respecto a su complejidad de peor caso?

Seleccione una o más de una:

- ☒ a. No se puede afirmar nada con respecto a la complejidad sin conocer el código del programa ✓
- ☐ b. Si tiene dos ciclos anidados, su complejidad es de $O(n^2)$
- ☐ c. No se puede decir con precisión qué complejidad tiene el programa, pero como tiene dos ciclos anidados podemos afirmar que es al menos $O(n^2)$ (podría ser $O(n^2)$, $O(n^3)$, ...)
- ☐ d. La complejidad del programa será bien $O(n^2)$ o $O(n^3)$, dependiendo si dentro de los ciclos el código es de complejidad lineal o no.

Respuesta correcta

La respuesta correcta es: No se puede afirmar nada con respecto a la complejidad sin conocer el código del programa

Pregunta **10**

Incorrecta

Puntúa 0 sobre 5

Para un vector v , decimos que un **valle** es un subarreglo de valores que cumple que

- todos los valores del subarreglo son iguales
- Los vecinos inmediatos del subarreglo (si existen) son de valor mayor estricto que los valores del subarreglo.



El siguiente programa calcula **el valle más largo** de un vector.

```
int valleMasLargo(vector<int> &v) {
    int i = 0;
    int maxValle = 0;
    int valle;
    while (i < v.size()) {
        int j = i + 1;
        while (j < v.size() && v[i] == v[j]) {
            j++;
        }
        if ( (i == 0 || v[i] < v[i-1])
            && (j == v.size() || v[j] > v[j-1]) ) {
            valle = j - i;
        }
        i = j;
        if (valle > maxValle) {
            maxValle = valle;
        }
    }
    return maxValle;
}
```

Seleccione su complejidad de peor caso en función de n donde $n = |v|$.

Si además este problema se puede resolver en una complejidad mejor, también seleccione la mejor complejidad de la lista.

Seleccione una o más de una:

- ☒ a. $O((\log n)^2)$  $O(n^2)$
- ☐ b. $O(1)$
- ☐ c. $O(n)$
- ☒ d. $O(n \log n)$  $O(\log n)$

Respuesta incorrecta.

La respuesta correcta es: $O(n)$

Pregunta **11**

Correcta

Puntúa 5 sobre 5

Se tiene dos secuencias ordenadas de enteros S_1 y S_2 de longitud N y M respectivamente.

A priori no se conoce nada más de las secuencias aparte de que están ordenadas.

Se desea calcular una secuencia S_3 que contenga los elementos presentes en ambas secuencias.

La cota más ajustada de complejidad que se puede conseguir para resolver este problema es de...

Seleccione una:

- ☐ a. $O(M \log N)$
- ☐ b. $O(N \log M)$
- ☒ c. $O(N+M)$ ✓
- ☐ d. $O(N*M)$

Respuesta correcta

La respuesta correcta es: $O(N+M)$

Pregunta **12**

Correcta

Puntúa 5 sobre 5

Un sitio de comercio electrónico recibe **a lo largo del tiempo** pedidos de compra a través de su página web.

En lugar de atender los pedidos por orden de llegada, deciden atender primero los pedidos de mayor monto total.

Para ello, quieren mantener su lista de pedidos en orden decreciente de valor para poder elegir rápidamente los pedidos a atender.

Nos consultan a nosotros qué estrategia de ordenamiento adaptaríamos al problema para minimizar el tiempo de procesamiento de cada paquete nuevo. Nosotros les recomendamos...

Seleccione una:

- ☒ a. Insertion Sort ✓
- ☐ b. Selection Sort
- ☐ c. Bubble Sort
- ☐ d. Merge Sort

Respuesta correcta

La respuesta correcta es: Insertion Sort

Pregunta **13**

Sin contestar

Sin calificar

Escriba abajo cualquier comentario que le parezca útil y/o relevante sobre el simulacro y su desarrollo.

.....