

## ISIS 2103 Diseño de Algoritmos

Semestre 2015-1

Tarea No. 1

Prof. Rodrigo Cardoso

Fecha de entrega: Febrero 23, 2015, 8:30 (por Sicua+)

---

1 [30/100] El lenguaje GCL, elegido para expresar los algoritmos en el curso, tiene peculiaridades que no son corrientes en los lenguajes de programación comerciales. Dé respuestas para las siguientes preguntas:

1a Suponga que GCL se enriquece con una instrucción nueva  $S$ , pero que ésta se puede implementar con instrucciones GCL ya conocidas. ¿Es factible enriquecer el cálculo de Hoare con una regla que permita concluir la corrección de  $S$  con respecto a una especificación dada?

1b Dados dos programas,  $S_1$  y  $S_2$ , se dice que  $S_1$  *simula*  $S_2$  cuando, para toda especificación  $\langle Q, R \rangle$  (pre / poscondición) se tiene que

$$\{Q\} S_2 \{R\} \Rightarrow \{Q\} S_1 \{R\}$$

Defina la relación

$$S_1 \text{ equivale } S_2 \equiv (S_1 \text{ simula } S_2) \wedge (S_2 \text{ simula } S_1)$$

Muestre que *equivale* es una relación de equivalencia. Explique, en términos operacionales, cuándo dos programas son equivalentes.

---

2 [30/100] Dadas las funciones de variable real positiva

$\log [n]$	$3^{\log n}$	$n^\pi$
$n!$	$(e/2)^n$	$(e/3)^n$
$(+k \mid 1 \leq k \leq n: k/2^k)$		$(*k \mid 1 \leq k \leq \log [n]: k/2^k)$

Ordénalas en una secuencia  $f_1, f_2, \dots, f_8$ , de manera que  $f_i = O(f_{i+1})$ , para  $i=1, 2, \dots, 8$ .

---

3 [40/100] Dada una matriz de enteros  $A[0..m-1, 0..n-1]$ , donde cada fila está ordenada ascendentemente, y un entero  $x: \text{int}$ , se quiere saber si  $x \in A$ . Para esto se usa la función *busa*, abajo descrita; a su vez, dentro de su cuerpo, *busa* llama a la función *busbin*:

```
funct busbin (b[0..n-1]: int; p,q:int, x:int):int
{Pre Q:  ( $\forall k \mid 0 \leq k < n-1: b[k] \leq b[k+1]$ )  $\wedge p \leq q-1$ }
{Pos R:  ( $\text{busbin}=n \wedge x \notin b[0..n-1]$ )  $\vee (0 \leq \text{busbin} < n \wedge x=b[\text{busbin}])$ }
[ if p=q-1
  then if b[p]=x then busbin:= p
        else busbin:= n
    fi
  else r:= (p+q)  $\div$  2;
        if b[r]<x then busbin:= busbin(b,p,r)
        else busbin:= busbin(b,r+1,q)
    fi
fi
]
```

```

funct busa (A[0..m-1,0..n-1]: int; x: int): boolean
{Pre Q:  (∀i | 0 ≤ i < m: (∀j | 0 ≤ j < n: A[i,j] ≤ A[i,j+1]))}
{Pos R:  (¬busa ∧ x ∉ A) ∨ (busa ∧ x = A[i,j])}
}
[ i:= 0;
  j:= n;
  do i ≠ m ∧ j = n    →      j:= busbin(A[i,.],0,n)
                        if j ≠ n      → skip
                        [] j = n      → i:= i+1
                        fi
  od;
  busa:= (i ≠ m);
]

```

Para el problema de calcular  $\text{busa}(A[0..m-1,0..n-1],x)$  con el algoritmo anterior, defina como operación básica la asignación de variables.

- 3a** ¿Cuál es el tamaño del problema? (i.e., de qué variable(s) depende el tamaño del problema; o bien, defina el tamaño en función de variables de la llamada).
- 3b** Estime el peor caso del tiempo  $T_{\text{busa}}$  (como  $\theta(\dots)$ ).
- 3c** ¿Cuándo se presenta el peor caso?