ISIS 1105 Diseño y Análisis de Algoritmos **Nombre: ___________________________**
Semestre 2013-10 – Parcial 1 **Sección: 3**
Febrero 21, 2012
Prof. Rodrigo Cardoso

En algunos de los puntos siguientes se usa una notación de procedimientos para extender el lenguaje GCL. Un procedimiento p, con parámetros **x** (un vector de parámetros) se declara así:

```
proc p(x)
 Ctx C
 Pre Q
 Pos R
 ⌈ ⟨cuerpo del procedimiento⟩ ⌋
```

y se llama con una instrucción de la forma (**a** es un vector de argumentos):

```
p(a)
```

La semántica de las llamadas de procedimientos es la usalmente entendida. La especificación `[Ctx, Pre, Pos]` es parte de la documentación y no es sintácticamente obligatoria.

1 **(20 puntos)** Se define que dos programas `S1`, `S2` son *equivalentes*, y para ello se usa la notación infija `.≅.`, si cumplen que:

```
S1 ≅ S2  ≡  (∀R |: wp(S1|R) ≡ wp(S2|R))
```

1a (15/20) Pruebe que:

```
S1 ≅ S2  ≡  (∀Q,R |: {Q} S1 {R} ≡ {Q} S2 {R})
```

1b (5/20) Explique qué quiere decir, en términos de semántica operacional, el resultado **1a**.

2 **(40 puntos)** Considere el siguiente programa GCL, que permuta los elementos de un arreglo de elementos de tipo `X`. Se sabe que `(X,≤)` es un tipo ordenado que admite instrucciones de comparación en los algoritmos. El programa funciona de modo que, al final, todos los elementos menores o iguales a un valor `x:X` están en los primeros índices del arreglo:

```
[ Ctx C: b:[0..n-1] of X  ∧  n:nat  ∧  x:X
 {Pre Q: true}

  p:= 0;
  q:= n;

 {Inv P: 0≤p≤n  ∧  0≤q≤n  ∧  b[0..p-1]≤x  ∧  b[q..n-1]>x}

  do p≠q      →    if   b[p]≤x     →    p:= p+1;
                   []   b[p]>x     →    q:=q-1;
                                        temp:= b[p];
                                        b[q]:= b[p];
                                        b[p]:= temp
                   fi
  od

 {Pos R: 0≤p<n  ∧  b[0..p-1]≤x  ∧  b[p..n-1]>x}
]
```

2a (25/40) Sea $T_2(n)$ el tiempo, en el peor caso, para la ejecución del programa, para $n \geq 0$. Calcule T_2 de manera exacta. Para esto, considere como operaciones básicas

(i) las asignaciones a variables enteras, con costo `1`.

(ii) las asignaciones a variables de tipo `X`, con costo m^2.

2b (10/40) Suponga que el valor `m` es tal que $m = \theta(\sqrt{n})$. Estime $T_2(n)$ como $\theta(f(n))$ para alguna función `f`.

2c (5/40) Estime $S_2(n)$, el uso de espacio en peor caso, $\theta(f(n))$ para alguna función `f`. Suponga que las variables enteras cuestan `1` y las de tipo `X` cuestan m^2. Suponga que $m = \theta(\sqrt{n})$.

3 **(40 puntos)** Supóngase un procedimiento recursivo cuya semántica depende de un parámetro entero, que tiene la siguiente propiedad (los b's son predicados y los c's y g's son procedimientos):

```
proc p(n: nat)
⌈ if  n=0   → skip
  []  n>0   → if b(n)   → c(n)
               [] ¬b(n) → p(n-1); g(n)
               fi
  fi
⌋
```

Se quiere estimar el peor caso del tiempo de ejecución de `p(n)`. Como operación básica tómense las llamadas a `g`, y supóngase que la llamada a `g(n)` tiene costo $n*2^n$, para $n>0$.

3a (15/40) Sea $T_3(n)$ el tiempo, en el peor caso, para la ejecución de `p(n)`, $n \geq 0$. Establezca una relación de recurrencia que defina T_3.

3b (25/40) Estime $T_3(n)$ como $\theta(f(n))$ para alguna función `f`.