

Carnet: _____

Nombre: _____

Quiz
(20 puntos)

El quiz consta de 2 (DOS) preguntas. Revíselo bien antes de empezar.

Pregunta 0	Pregunta 1	Total
5 puntos	15 puntos	20 puntos

(Espacio adicional)

Pregunta 0 — 5 puntos

Demuestre formalmente que toda función f asintóticamente no-negativa no se tiene a sí misma como cota superior estricta, esto es, que $f(n) \neq o(f(n))$.

Nota: No confunda la notación o con O (esto es, no confunda o -pequeña con O -grande).

Otra nota: Debe usar la definición formal de o presentada en el libro de texto de Cormen et al.

Pregunta 1 — 15 puntos

Se desea construir un programa eficiente (esto es, asintóticamente eficiente) que determine la inclusión de un arreglo en otro, de acuerdo con la siguiente especificación:

```
|| const  $N : int$  ;  
     $a0, a1 : array [0 .. N)$  of  $T$  ;  
var  $incl : boolean$  ;  
  
    {  $N \geq 0$  }  
  
    DETERMINARINCLUSIÓN  
  
    {  $incl \equiv (\forall i : 0 \leq i < N : (\exists j : 0 \leq j < N : a0[i] = a1[j]))$  }  
  
||
```

donde T es un tipo que cuenta con una relación de orden total (también llamada orden lineal).

Para la realización del programa, Ud. cuenta con varios procedimientos ya construidos $busqL$, $busqB$, $ordI$, $ordS$, $ordB$ y $ordM$, que implementan adecuadamente búsqueda y ordenamiento en arreglos. Las interfaces de sus encabezados corresponden a

```
proc  $busq$  (in  $N : int$  ; in  $a : array [0 .. N)$  of  $T$  ; in  $e : T$  ; out  $b : boolean$ )
```

y

```
proc  $ord$  (in  $N : int$  ; in-out  $a : array [0 .. N)$  of  $T$ ) .
```

Los procedimientos $busqL$ y $busqB$ cuentan con el encabezado de $busq$ e implementan respectivamente búsqueda lineal y búsqueda binaria, mientras los procedimientos $ordI$, $ordS$, $ordB$ y $ordM$ cuentan con el encabezado de ord e implementan respectivamente los algoritmos de ordenamiento por inserción, selección, burbuja y mezcla.

Se desea entonces que Ud. haga lo siguiente:

- (a) Construya un programa eficiente (asintóticamente eficiente) que resuelva el problema DETERMINARINCLUSIÓN, haciendo uso de los procedimientos disponibles mencionados.

Su programa debe ser lo más eficiente posible en términos de tiempo de ejecución. No se exige eficiencia en términos de espacio de memoria, por lo cual podrá utilizar arreglos auxiliares (notar que la especificación exige que los arreglos de entrada permanezcan constantes).

Su solución debe ser iterativa. Puede utilizar cuantas iteraciones quiera, pero no puede utilizar recursión. En cada una de sus iteraciones Ud. deberá indicar el invariante y la función de cota asociadas (aunque, por supuesto, no se requiere la demostración de correctitud).

- (b) Dé el orden de complejidad del tiempo de ejecución del peor caso de su solución al problema DETERMINARINCLUSIÓN. Indique muy brevemente cómo obtuvo el orden, sin necesidad de justificar muy detalladamente su respuesta.

(Espacio adicional)

(Espacio adicional)