

Carnet: _____

Nombre: _____

Examen I

(30 puntos)

Antes de empezar, revise bien el examen, el cual consta de 3 (tres) preguntas.

Pregunta 0	Pregunta 1	Pregunta 2	Total
6 puntos	10 puntos	14 puntos	30 puntos

Pregunta 0 - 6 puntos

Considere el tipo abstracto de datos *Diccionario* ($T0, T1$) y considere la especificación de éste con modelo abstracto presentada tanto en clase como en el material de apoyo del curso. El modelo de representación en cuestión cuenta con un atributo correspondiente a la capacidad máxima de almacenamiento del diccionario, un segundo atributo con el conjunto de claves que el diccionario conoce, y un tercer y último atributo con la función de correspondencia entre claves y valores que el diccionario almacena:

```
const max : int  
var conoc : set (T0)  
    tabla : T0 → T1      ,
```

y estos atributos están restringidos por el siguiente invariante de representación:

$$\text{max} > 0 \wedge \# \text{conoc} \leq \text{max} \wedge \text{conoc} = \text{dom}(\text{tabla}) \quad .$$

Se desea que Ud. especifique formalmente un par de operaciones de este tipo abstracto de datos en términos de este modelo.

(0.0) Especifique una operación con interfaz

```
proc agregar ( in-out d : Diccionario ; in c : T0 ; in v : T1 )
```

que permita agregar el par de asociación entre la clave c y el valor v al diccionario d , permitiendo que c pueda ser o no ser una clave preexistente en d al momento de inicio de la operación.

(0.1) Especifique una operación con interfaz

```
proc valorPresente ( in d : Diccionario ; in v : T1 ; out n : int ; out c : T0 )
```

que determine si el valor v está presente en el diccionario d . El parámetro de salida n indicará la cantidad de claves que están asociadas a v en d y, en caso de haber al menos una de tales claves, c contendrá una de ellas.

Pregunta 1 - 10 puntos

Un par de estudiantes, Yajaira y Alfonso, decidieron cambiar el algoritmo tradicional de ordenamiento por mezcla dividiendo los segmentos a ordenar en tres subsegmentos en lugar de sólo dos subsegmentos. Propusieron entonces la solución, parcialmente especificada, presentada a continuación:

```
proc ordMezcla (in n : int ; in-out a : array [0..n] of T ; in inf, sup : int)
  { Pre : 0 ≤ inf ≤ sup ≤ n ∧ ... }
  { Post : ... }
  { Cota variante : sup - inf }
  [
    if sup - inf ≤ 1 → skip
    [ sup - inf ≥ 2 → [ var m0, m1 : int ;
                        m0, m1 := [?], [?]
                        { inf < m0 ≤ m1 < sup }
                        ; ordMezcla (n, a, inf, m0)
                        ; ordMezcla (n, a, m0, m1)
                        ; ordMezcla (n, a, m1, sup)
                        ; mezclar3 (n, a, inf, m0, m1, sup)
                    ]
    ]
  fi
],
```

con el cual se puede ordenar una secuencia s almacenada en un arreglo de tamaño m mediante una llamada principal de la forma $ordMezcla(m, s, 0, m)$.

Responda ahora las siguientes preguntas:

- (1.0) Para el diseño de las expresiones a utilizar en la asignación a las variables $m0$ y $m1$, Yajaira y Alfonso no estaban seguros de qué aserción intermedia debían utilizar como postcondición de tal asignación y una profesora les sugirió utilizar la aserción arriba indicada. Antes de consultar con la profesora en cuestión, Yajaira había propuesto la aserción $inf \leq m0 \leq m1 \leq sup$ y Alfonso había propuesto $inf < m0 < m1 < sup$.

Explique, brevemente pero de manera clara y completa, por qué las propuestas de Yajaira y de Alfonso no habrían permitido garantizar que el programa funcionase correctamente.

- (1.1) Suponga que $m0$ y $m1$ son escogidos de forma tal que los tamaños de los tres subsegmentos siempre redondean la tercera parte del tamaño del segmento de entrada (sea con piso $\lfloor \dots / 3 \rfloor$ o sea con techo $\lceil \dots / 3 \rceil$), y considere además que el tiempo de ejecución del algoritmo para mezclar tres subsegmentos tiene el mismo orden de complejidad que el algoritmo tradicional de dos subsegmentos, esto es, complejidad lineal sobre el tamaño del segmento total de entrada.

Especifique la relación de recurrencia correspondiente a la complejidad del tiempo de ejecución de esta versión de ordenamiento por mezcla, y use el Teorema Maestro para resolverla e indicar si Yajaira y Alfonso lograron mejorar el orden de complejidad asintótico del tiempo de ejecución del ordenamiento por mezcla original.

- (1.2) Por último, Alfonso le insiste a Yajaira en volver al esquema de mezclar sólo dos subsegmentos y cambia el código posterior a la aserción intermedia a

```
    ; ordMezcla (n, a, inf, m0) ; ordMezcla (n, a, m0, sup) ; mezclar2 (n, a, inf, m0, sup)
```

correspondiendo $mezclar2$ a la tradicional mezcla de sólo dos subsegmentos, pero Alfonso mantiene la asignación anterior a $m0$ y $m1$ que permitía obtener tres subsegmentos de tamaños que redondeaban la tercera parte del tamaño del segmento de entrada.

Especifique la nueva relación de recurrencia correspondiente a la complejidad del tiempo de ejecución de esta nueva versión de Alfonso del ordenamiento, y explique por qué ahora el Teorema Maestro no podría ser usado para resolver la nueva recurrencia. Use entonces el método de árbol de recursión para proponer una cota inferior y una cota superior para la complejidad del tiempo de ejecución de esta nueva versión del ordenamiento.

Pregunta 2 - 14 puntos

Se desea que Ud. construya un procedimiento que implemente una cierta versión del algoritmo de ordenamiento por selección, según los lineamientos que se señalan a continuación.

Usaremos la siguiente especificación, que corresponde a la convencional utilizada en clase y presentada en el material de apoyo del curso:

```
proc seleccion (in  $n : int$ ; in-out  $a : array [0..n]$  of  $T$ )  
  { Pre :  $n \geq 0 \quad \wedge \quad mcjto(a, 0, n) = M$  }  
  { Post :  $ord(a, 0, n) \wedge mcjto(a, 0, n) = M$  }
```

En esta especificación se utiliza la función auxiliar *mcjto*, la cual permite obtener el multiconjunto de elementos almacenados en algún segmento de un arreglo, cuya definición corresponde a

$$mcjto(x, y, z) = \llbracket i : y \leq i < z : x[i] \rrbracket .$$

La variable de especificación M se refiere al multiconjunto del segmento señalado en la precondición, el cual debe permanecer igual al final del procedimiento. También se utiliza el predicado auxiliar *ord*, el cual señala que algún segmento de un arreglo está ordenado y cuya definición corresponde a

$$ord(x, y, z) \equiv (\forall i, j : y \leq i \leq j < z : x[i] \sqsubseteq x[j]) ,$$

donde \sqsubseteq denota la relación de orden utilizada sobre los elementos de tipo T almacenados en el arreglo.

Se desea que Ud. dé una implementación iterativa de *seleccion*, cuya iteración principal debe ser construida según el siguiente invariante, especificado parcialmente:

$$0 \leq k \leq n \wedge ord(a, k, n) \wedge \dots ,$$

siendo k una nueva variable entera.

De acuerdo con todo lo señalado, responda lo siguiente:

- (2.0) Complete la implementación utilizando iteración (no puede utilizar recursión), señalando tanto el invariante completo como la cota variante de terminación correspondientes a todas las iteraciones que utilice.
- (2.1) Demuestre que el final de su iteración principal se comporta correctamente, demostrando que su invariante garantiza la postcondición al finalizar.