



**Estructuras de Datos**  
**Evaluación nro. 1 (todas las coordinaciones)**

Nombre: \_\_\_\_\_ RUT: \_\_\_\_\_ Sección: \_\_\_\_\_

Tiempo de desarrollo: 80 minutos

30 de mayo de 2023

**Pregunta 1 (15 puntos):**

Dado el arreglo **A:[33, 44, 11, 77, 23, 65]**, y **n** el largo del arreglo, muestre la traza al aplicar el algoritmo que se indica a continuación. Para realizar la traza use la tabla (lado derecho del algoritmo) para mostrar los cambios del arreglo A y de las variables **intervalo**, **i** y **j** en cada iteración.

**algoritmo\_preg1** (num A[], num n)

num intervalo, i, j

intervalo = n//2

WHILE (intervalo > 0) DO

FOR i= intervalo TO n-1 DO

j = i - intervalo

WHILE (j >= 0) DO

IF ( A[j] > A[j+intervalo] ) THEN

swap( A[j], A[j+intervalo] )

j = j - intervalo

ELSE

j = -1

END-IF

END-WHILE

END-FOR

intervalo = intervalo//2

END-WHILE

Arreglo	intervalo	i	j
---------	-----------	---	---

**Nota:** Considere que // es la división entera (sin redondeo)

**Criterios de evaluación pregunta 1:**

*Item I: (3 puntos) La traza corresponde al algoritmo propuesto y utiliza el formato indicado.*

*Item II: (12 puntos) Los valores de la traza son correctos y completos de acuerdo a lo solicitado*



**Pregunta 2 (25 puntos):**

Dadas dos listas L1 y L2 de enteros, construya un algoritmo en pseudocódigo que duplique en L1 cada elemento de la lista que aparece además en L2. Debe definir claramente el tipo de lista usada y la estructura de datos correspondiente.

Ejemplo 1:

Sean L1 -> 2 -> 6 -> 3 -> 8 -> 4 -> 1 -> nulo y  
L2 -> 2 -> 1 -> 10 -> 4 -> 3 -> nulo  
Su algoritmo debe generar y retornar la lista  
L1 -> 2 -> 2 -> 6 -> 3 -> 3 -> 8 -> 4 -> 4 -> 1 -> 1 -> nulo

Describa las estructuras de datos utilizadas para resolver este problema indicando claramente por qué y para qué ha decidido definirlas de esa forma. NO asuma operaciones implementadas. (sólo existen las de creación, por ejemplo Nodo CrearNodo(dato) )

**Criterios de evaluación:**

*Item I: (5 puntos) El algoritmo propuesto apunta a resolver el problema planteado y define y argumenta la(s) estructura de datos requeridas para su solución.*

*Item II: (15 puntos) Cumplido total o parcialmente el ítem I, los algoritmos resuelven correctamente el problema planteado en consistencia con la estructura (s) definidas en el ítem I.*

*Item III: (5 puntos) Cumplido total o parcialmente el ítem I, el algoritmo está escrito en pseudocódigo ordenado y consistente, identificando correctamente entradas y salidas de los algoritmos.*

**Si no define las estructuras de datos solicitadas en el ítem I, los ítems II y III no serán evaluados**



**Pregunta 3 (25 puntos):**

Se le solicita escribir un algoritmo en pseudocódigo que **dada una pila P** de números unos y dos, y un **valor entero m**, determine si los elementos de P cumplen con la forma  $SS'S$ , donde **S** representa los primeros **m** elementos de P y **S'** representa los elementos invertidos de **S**.

Ejemplos:

Dada la pila P -> 2 -> 1 -> 1 -> 1 -> 1 -> 2 -> 2 -> 1 -> 1 -> nulo y m = 3

Su programa debe retornar **True** ya que al obtener S=211, los elementos de P cumplen la forma **211112211**

Dada P -> 2 -> 1 -> 1 -> 1 -> 1 -> 2 -> 2 -> 1 -> 2 -> nulo y m = 3

Su programa debe retornar **False** ya que al obtener S=211, los elementos de P no cumplen la forma **211112211**

Defina las estructuras de datos utilizadas para resolver este problema indicando claramente por qué y para qué ha decidido definirlas de esa forma. Asuma que sólo las operaciones elementales de TDA Cola y TDA Pila están implementadas. Puede usar pilas y colas auxiliares.

**Criterios de evaluación pregunta 3:**

*Item I: (5 puntos) El algoritmo propuesto apunta a resolver el problema planteado y define y argumenta la(s) estructura de datos requeridas para su solución.*

*Item II: (15 puntos) Cumplido total o parcialmente el ítem I, los algoritmos resuelven correctamente el problema planteado en consistencia con la estructura (s) definidas en el ítem I.*

*Item III: (5 puntos) Cumplido total o parcialmente el ítem I, el algoritmo está escrito en pseudocódigo ordenado y consistente, identificando correctamente entradas y salidas de los algoritmos.*

**Si no define las estructuras de datos solicitadas en el ítem I, los ítems II y III no serán evaluados**



Departamento de Ingeniería Informática  
Universidad de Santiago de Chile