

## PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA

### ALGORITMIA 4ta. práctica (tipo B) (Primer Semestre 2019)

#### Indicaciones Generales:

- Duración: 2h 50 min.
- Al inicio de cada programa, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- Si la implementación es significativamente diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- El orden será parte de la evaluación.
- Su trabajo deberá ser subido a PAIDEIA en el espacio indicado por los jefes de práctica.

#### **Pregunta 1 (10 puntos)**

Un almacén desea controlar el stock de sus productos, por tal motivo ha decidido utilizar un árbol binario de búsqueda. Para esta tarea decide utilizar nodos que almacenen el número del lote de producción, identificado por el año, mes y día (aaaammdd), y la cantidad de materiales que contiene cada lote. Además la política de la empresa es de despachar primero los productos que tienen mayor antigüedad, por tal motivo el ABB debe estar ordenado por el identificador de lote.

- a) Desarrolle una función recursiva que inserte un nuevo nodo en el árbol (3.5 puntos)
- b) Desarrolle una función recursiva que busque y obtenga el nodo más antiguo de un árbol (1 punto)
- c) Desarrolle una función recursiva que despache los productos del almacén de acuerdo a un pedido solicitado (5.0 puntos)

Por ejemplo si el árbol cuenta con los siguientes nodos:

(20170620 – 20) (20170810 – 20) (20180211 – 20) (20180409 – 10)

Y se solicita despachar 35 productos. El árbol resultante tendrá los siguientes nodos:

(20170810 – 5) (20180211 – 20) (20180409 – 10)

- d) Desarrolle una función recursiva que muestre en orden los lotes y cantidades que existen en el ABB que representa el almacén (0.5 puntos)

#### **Pregunta 2 (10 puntos)**

ISO 8583 es un estándar para transacciones financieras y define un formato de mensaje y un flujo de comunicación para que diferentes sistemas puedan intercambiar estas transacciones. Las transacciones incluyen compras, extracciones, depósitos, reintegros, reversos, consultas de saldo, pagos, transferencias entre cuentas y bloqueos de tarjetas o cuentas. Por otro lado, ALG 6969 es un estándar simplificado de la ISO 8583 desarrollado en el curso de Algoritmia de la PUCP. A continuación se describe el estándar ALG 6969:

Message Type Indicator (MTI) - Indicador de Tipo de Mensaje	Bitmap - Mapa de Bits	Data Elements - Campos de datos
--	-----------------------	---------------------------------

MTI: Este es un campo numérico de 4 dígitos que clasifica la función de alto nivel del mensaje. Un MTI incluye la versión ALG 6969, la clase (Message Class), la función (Message Function) y el origen del mensaje (Message Origin). A continuación se presentan un par de ejemplos:

Ejemplo 1 de MTI	Ejemplo 2 de MTI
2xxx ----> versión de ALG 6969 ( <b>Versión 2019</b> ) x2xx ----> clase de mensaje ( <b>Financiero</b> ) xx0x ----> función del mensaje ( <b>Petición</b> ) xxx0 ----> quien empezó la comunicación ( <b>usuario tarjeta habiente</b> )	2xxx ----> versión de ALG 6969 ( <b>Versión 2019</b> ) X9xx ----> clase de mensaje ( <b>Bloqueo</b> ) xx0x ----> función del mensaje ( <b>Petición</b> ) xxx2 ----> quien empezó la comunicación ( <b>usuario empleado de entidad bancaria</b> )

Bitmap: Dentro de ALG 6969, un mapa de bits es un campo o subcampo dentro de un mensaje que indica qué otros elementos (campos o subcampos) se encuentran en el mensaje. Este *Mapa de Bits* indica qué campos (Data Elements) del 1 al 8 están presentes. El mapa de bits se puede transmitir como un campo de 2 caracteres hexadecimales 0-9, A-F. Por ejemplo, el byte 42 en binario es '0100 0010' lo que significa que los campos 2 y 7 están presentes en este mensaje.

Data Elements: Son los campos individuales que llevan la información sustancial acerca de la transacción. Hay 8 campos definidos en ALG 6969:2019.

Data element	Tipo	Uso
1	n 2 (numérico de 2 dígitos)	Moneda. 01 es dólares, 02 es soles.
2	n 20	Número de cuenta origen o número de tarjeta. Si se trata de bloqueo, será la cuenta o tarjeta a bloquear.
3	n 6	Reservado, completados con ceros.
4	n 12	Monto de la transacción, siendo los dos últimos dígitos la parte decimal.
5	n 20	Número de cuenta destino.
6	n 12	Reservado, completados con ceros.
7	n 10	Fecha y hora de la transmisión en MMDDhhmmss
8	n 8	Reservado, completados con ceros.

Como ejemplo, analizaremos la siguiente trama en formato ALG 6969:

[illegible]

MTI (Tamaño: 4 dígitos)	2902	Versión ALG 2019   Bloqueo   Petición   Usuario empleado de la entidad bancaria
Bitmap (Tamaño: 2 caracteres)	42	Significa que los campos 2 y 7 están presentes en este mensaje.
Data Elements (Tamaño 90 dígitos)	00 (Campo 1) 12345123451234512345 (Campo 2) 000000 (Campo 3) 0000000000000 (Campo 4) 00000000000000000000 (Campo 5) 0000000000000 (Campo 6) 0423143052 (Campo 7) 00000000 (Campo 8)	Moneda: ninguna Número de cuenta de origen Reservado Monto de la transacción: S/. 0.0 Número de cuenta destino Reservado Fecha y hora de la transmisión: 23 de abril a las 14:30:52 Reservado

El Banco Decrépito ha decidido utilizar el estándar simplificado ALG 6969 para sus transacciones financieras. Es por ello que ha implementado un árbol de búsqueda binaria (ABB) en el servidor central del banco para recibir todas las tramas de peticiones de operaciones financieras enviadas desde los diferentes terminales desde cualquier parte del mundo. Cada nodo del árbol estará conformado por la trama completa de una operación y los dos punteros a sus hijos, izquierdo y derecho. La forma como se agrega un nodo al ABB será considerando el monto de la operación. Si es el primer nodo, será el nodo raíz. Si es un nodo con un monto igual a la operación de otro nodo del ABB, el nuevo nodo será adicionado como un hijo izquierdo del nodo cuyo monto es igual al de la operación del nuevo nodo.

Luego de haber añadido todas las operaciones como nodos en el ABB, deberá mostrar las operaciones de los nodos del ABB **en orden de manera iterativa**.

Elabore un programa en C que haga lo siguiente:

- a) Defina la estructura para el ABB de operaciones o tramas AL6969 del banco (1 punto).
- b) Defina la función que permita el ingreso de una operación en formato ALG 6969 al ABB (3 puntos).
- c) Defina la función que permita mostrar en pantalla las operaciones en formato ALG6969 del ABB. Cada operación deberá estar en una línea. Recuerde que el recorrido es **en orden y de manera iterativa** (5 puntos).
- d) Defina la función main que permita armar todo el árbol con las operaciones del arreglo de tramas ALG6969, para luego mostrar en pantalla las operaciones del árbol invocando a la función de la parte (c) (1 punto).

Profesores del curso:     Rony Cueva  
                                     Johan Baldeón

Pando, 28 de mayo del 2019