

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

**ALGORITMIA Y ESTRUCTURA DE DATOS**

**4ta. práctica (tipo B)**  
**(Primer Semestre 2024)**

Duración: 1h 50 min.

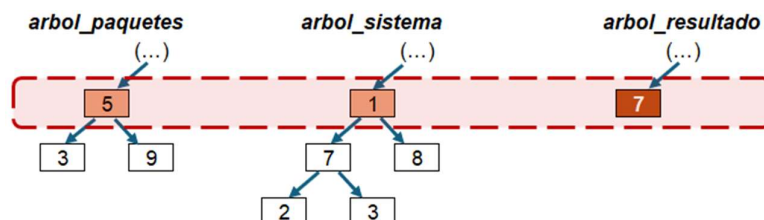
- **No puede utilizar apuntes, solo hojas sueltas en blanco.**
- En cada función el alumno deberá incluir, a modo de comentario, la forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear plantillas o funciones no vistas en los cursos de programación de la especialidad.
- Los programas deben ser desarrollados en el lenguaje C++. Si la implementación es 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.
- **Solo está permitido acceder a la plataforma de PAIDEIA, cualquier tipo de navegación, búsqueda o uso de herramientas de comunicación se considera plagio por tal motivo se anulará la evaluación y se procederá con las medidas disciplinarias dispuestas por la FCI.**
- Para esta evaluación solo se permite el uso de las librerías `iostream`, `iomanip`, `climits`, `cstring`, `cmath` o `fstream`
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans.**
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_LAB4_P#` (donde # representa el número de la pregunta a resolver)

---

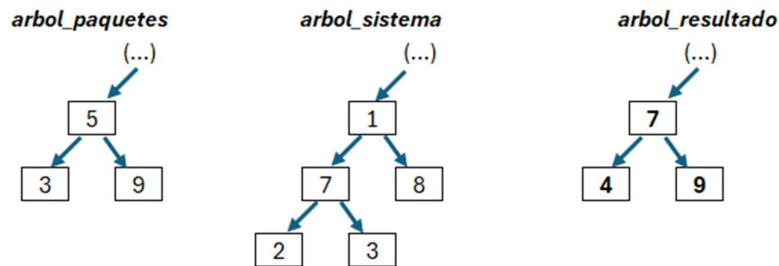
**Pregunta 1 (10 puntos)**

La detección de anomalías consiste en identificar acciones, eventos inusuales o no esperados en una tecnología (sistema, red, dispositivo, etc). En seguridad de redes, se evalúa un conjunto de paquete de datos de la siguiente manera:

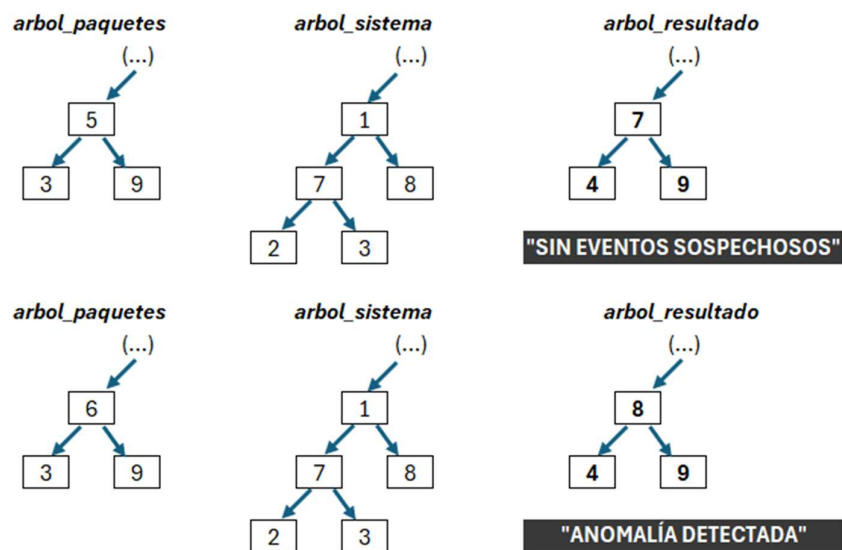
- Cada conjunto de paquetes de datos se representa con un ABB (**arbol\_paquetes**) donde a cada paquete de datos se le asigna un valor entero positivo.
- La herramienta de detección de anomalías genera un AB (**arbol\_sistema**) que será utilizado para el análisis de la posible anomalía
- El análisis se realiza nodo a nodo según el nivel en el que se encuentra. Al elemento del nodo del **arbol\_paquetes** se le suma el número de nodos que tiene el subárbol de **arbol\_sistema** y se le resta el número de hojas que tiene el subárbol de **arbol\_sistema**. Gráficamente es:



- El resultado de aplicar el arbol\_sistema al arbol\_paquetes resulta en un AB:



- Si el arbol\_resultado cumple con 2 condiciones: (1) es un ABB y (2) la suma de todos los nodos es un número par, se indica que no hay eventos sospechosos. Caso contrario, se indica que hay una anomalía detectada.



Se le solicita:

- Establecer la función principal, las estructuras de datos y demás funciones que permitan manejar lo descrito sobre la simulación de detección de anomalías. Debe construir los árboles: *arbol\_paquetes* y *arbol\_sistema*. Asimismo debe invocar a las funciones de b) y c). Como resultado final, el usuario debe de recibir un mensaje de "Anomalía detectada" o "Sin eventos sospechosos" según sea el caso. (2 puntos)
- Elaborar la función **aplicar\_arbol** que reciba como parámetros los árboles *arbol\_paquetes* y *arbol\_sistema* y que devuelva el *arbol\_resultado*. (4 puntos)
- Elaborar la función **determinar\_anomalia** que reciba el *arbol\_resultado* y permita verificar si es un ABB y si la suma de sus nodos es un número par (4 puntos)

## Pregunta 2 (10 puntos)

Un prestigioso equipo de editores ha recibido la tarea de organizar un libro de gran importancia que contiene numerosos capítulos y secciones. Este libro ha sido estructurado utilizando un árbol binario para facilitar su navegación y búsqueda. Sin embargo, encontrar el "Capítulo Principal" es crucial para entender la temática central del libro. A continuación, se presenta la información necesaria para esta tarea:

- El libro ha sido estructurado utilizando un árbol binario para organizar sus capítulos y secciones.
- El árbol binario tiene una altura igual a  $N$ , y no se conoce exactamente en qué nodo se encuentra el capítulo o sección de mayor relevancia.
- Cada nodo del árbol guarda el título del capítulo o sección y su nivel de relevancia en una escala del 1 al 10.
- El árbol guarda su tamaño total (número de nodos). Este tamaño se actualiza automáticamente al agregar o remover nodos.
- Se sabe que el capítulo o sección de mayor relevancia es el único con un nivel de relevancia de 10 y que está etiquetado como capítulo "Principal".

Se le pide que desarrolle lo siguiente:

a) Construya el árbol según la descripción siguiente (2 puntos).

### Descripción del Árbol:

- Raíz (Nivel 0):
  - Título: Título, Relevancia: 7
- Nivel 1:
  - Izquierda: Capítulo1, Relevancia: 8
  - Derecha: Capítulo2, Relevancia: 5
- Nivel 2:
  - Capítulo 1:
    - Izquierda: Sección1.1, Relevancia: 6
    - Derecha: Principal, Relevancia: 10
  - Capítulo 2:
    - Izquierda: Sección2.1, Relevancia: 3
    - Derecha: Sección2.2, Relevancia: 4
- Nivel 3:
  - Sección 1.1:
    - Izquierda: Sección1.1.1, Relevancia: 4
    - Derecha: Sección1.1.2, Relevancia: 2

b) Desarrolle una función **iterativa** que detecte si en el árbol binario se encuentra el "Capítulo Principal". Para esta labor, debe recorrer el árbol en **amplitud** (BFS, por sus siglas en inglés). Puede utilizar una estructura auxiliar ya sea una pila o una cola, en ella puede almacenar la información que necesite, con la excepción de arreglos. Debe dar como salida el nivel en el cual se encuentra el capítulo y mostrar la información del árbol por niveles separados con un cambio de línea (5 puntos).

### Salida de datos:

Título	7,			
Capítulo1	8, Capítulo2	5,		
Sección1.1	6, Principal	10, Sección2.1	3, Sección2.2	4,
Sección1.1.1	4, Sección1.1.2	2,		
El nivel es:	2			

- c) Se ha descubierto que los capítulos y secciones en la parte derecha del árbol contienen errores críticos y deben ser eliminados para mantener la coherencia del libro. Desarrolle una función que remueva la parte izquierda o derecha del árbol desde un nodo específico, según lo indicado. Asegúrese de que el tamaño del árbol se actualice correctamente después de esta operación (3 puntos).

**Salida de datos después de eliminar la parte derecha del árbol**

Título	7,	
Capítulo1	8,	
Sección1.1	6, Principal	10,
Sección1.1.1	4, Sección1.1.2	2

**Recuerde que cada nodo del árbol esta formado por un char titulo[50] y un int relevancia.**

Al finalizar el laboratorio, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para este laboratorio.

Profesores del curso:

Ana Roncal  
Fernando Huamán  
David Allasi  
Rony Cueva

San Miguel, 1 de junio del 2024