

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

**4ta. práctica (tipo B)**  
**(Segundo Semestre 2022)**

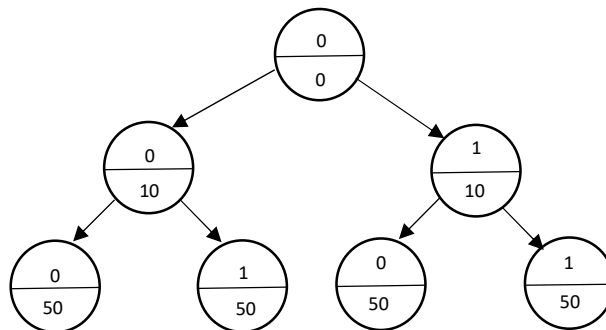
Duración: 2h 50 min.

- En cada función el alumno deberá incluir, a modo de comentario, la estrategia o forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- 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.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- Para este examen solo se permite el uso de las librerías **stdio.h**, **math.h**, **stdlib.h** y **string.h**
- Su trabajo deberá ser subido a PAIDEIA.
- Los archivos deben llevar como nombre su código de la siguiente forma **codigo\_LabX\_PY**

---

**Pregunta 1 (10 puntos)**

El empleo de cromosomas es una buena estrategia para problemas de asignación, usualmente los cromosomas son desarrollados empleando arreglos y convirtiendo las posibles asignaciones a números decimales, lo cual limita la cantidad de elementos al rango de números enteros. Por tal motivo la búsqueda de nuevas estructuras que no tengan restricciones con la cantidad de elementos es necesaria para problemas de mayor complejidad. Una estructura que puede apoyar a resolver este problema son los árboles binarios, los cuales pueden formar todas las posibles combinaciones que llevan a una solución al recorrer sus distintas ramas. A continuación, se muestra un ejemplo de como se puede representar un arreglo de 2 elementos, por ejemplo [10, 50]



De este ejemplo podemos obtener las siguientes combinaciones validas [0,0,1]->(50), [0,1,0]->(10) y [0,1,1]-> (10,50) que se obtienen de recorrer el árbol desde la raíz hasta las hojas que no tienen hijos.

**Se le pide:**

- a) Implemente una función que genere un árbol que soporte la estructura tipo cromosoma, similar a la que se muestra en la imagen anterior. Para la creación de esta estructura, se recibe un arreglo con los elementos a combinar, así como la cantidad de elementos del arreglo (5 puntos).
- b) Consideremos que el arreglo se trata de un conjunto de paquetes con diferentes pesos y se le solicita buscar cuantas combinaciones de paquetes cumplen con un peso total recibido. Por ejemplo, un arreglo de paquetes con los siguientes pesos [10,50,20,30,40], si se busca un peso total de 70 kg, pueden generarse 3 diferentes combinaciones que satisfacen lo solicitado [50,20],[30,40],[10,20,40]. Implemente una función que calcule el número de combinaciones que cumplen con el peso total brindado, para esta tarea debe usar el árbol generado en la pregunta anterior (5 puntos).

**Nota: No debe usar estructuras auxiliares para esta pregunta, solo puede usar un arreglo para recibir los pesos de los paquetes, pero no pueden modificarse.**

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

Una empresa controla el stock de sus productos, utilizando arboles binarios de búsqueda. Por tal motivo genera un ABB para cada uno de sus almacenes. 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. Luego de un tiempo de uso de esta solución, se decide unificar algunos almacenes, ya que manejan los mismos productos y muchas veces tienen los mismos lotes. Para esta tarea se debe considerar algunos detalles:

- Al unificar dos almacenes, se debe mover los productos al almacén que tiene la mayor cantidad lotes (nodos). El almacén que recibe los lotes se denominará **destino** y el almacén de donde se sacaran los lotes se le denomina **emisor**.
- Si en caso los 2 almacenes tienen la misma cantidad de lotes, se puede escoger cualquiera de ellos.
- Cuando un lote no existe en el árbol destino el mismo se inserta sin mayores complicaciones.
- Si el lote ya existe en el árbol destino, deben sumarse los stocks del lote y **no insertar nodos con el mismo lote**.
- El proceso de unificación debe realizarse de tal forma que se minimice la cantidad de cambios o movimientos de nodos en el almacén emisor. Por ejemplo, no se debe eliminar nodos que aún tienen hijos (hojas) al sacarlos del árbol emisor.

Por ejemplo, si el árbol destino cuenta con los siguientes nodos (inorden):

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

Y el árbol emisor tiene los siguientes nodos:

(20170811– 5) (20180211 – 10) (20180410 – 15)

Al final del proceso el árbol destino tendrá los siguientes nodos:

(20170620 – 20) (20170810 – 20) (20170811– 5) (20180211 – 30) (20180409 – 10) (20180410 – 15)

- a) Desarrolle las estructuras y funciones necesarias, para el ingreso adecuado de los datos en árbol (1.0 punto)
- b) Desarrolle una función que fusione los dos almacenes. Imprima los árboles en preorden y enorden antes y después de la fusión. Recuerde que no puede usar ninguna estructura auxiliar para realizar la operación (9.0 puntos)

Profesores del curso:

David Allasi  
Rony Cueva

San Miguel, 04 de noviembre del 2022