

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

4ta. práctica (tipo B)
(Segundo Semestre 2023)

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.
- No puede emplear STL, Plantillas o funciones no vistas en los cursos 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.
- 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.
- **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.**
- **Los programas deben ser desarrollados utilizando nombres para las funciones y variables en español, al igual que los comentarios. El uso de otro idioma anula su respuesta.**
- Para esta evaluación solo se permite el uso de las librerías **iostream, iomanip, climits cmath, fstream y cstring**
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans. Solo pueden usar una biblioteca para cada tipo de TAD, cualquier unión de bibliotecas invalidará su respuesta.**
- 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)

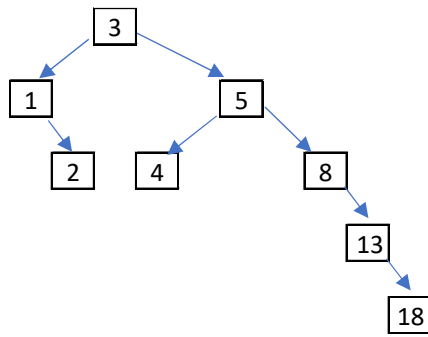
El Oficial de Seguridad de la Información de la empresa donde labora le solicita actualizar el método de autenticación de dos factores. Luego de solicitar primero al usuario sus credenciales (usuario y contraseña) como primer factor de autenticación, se le solicita que ingrese un grupo de números generados por token físico.

El método que usted plantea es el siguiente:

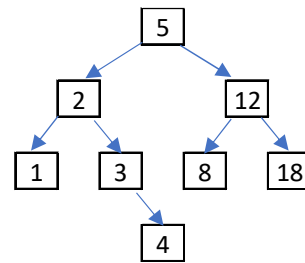
1. El usuario ingresa un grupo de 8 códigos (números enteros de 1 a 9), los cuáles pueden ser repetidos. Los códigos se insertan en un árbol binario de búsqueda verificando que, si ya existe el código en el ABB, se generará un nuevo código que resulta de la suma del código inicial con el nodo mayor en el momento de la inserción. Luego el árbol de búsqueda binario generado, pasa por un proceso de balanceo. El árbol resultante es llamado **arbol_token**

Ejemplo: 2, 5, 2, 1, 5, 6, 3, 4 (códigos generados por el token físico)

Árbol generado

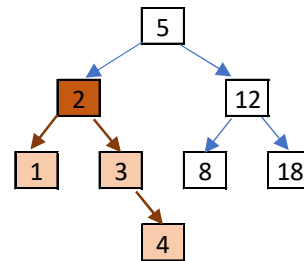


Árbol balanceado (*árbol_token*)



2. El sistema genera 2 números enteros que se utilizan para encontrar el *primer ancestro común* que tengan en el ABB balanceado (resultado del paso 1)

Números generados por el sistema: 1, 4

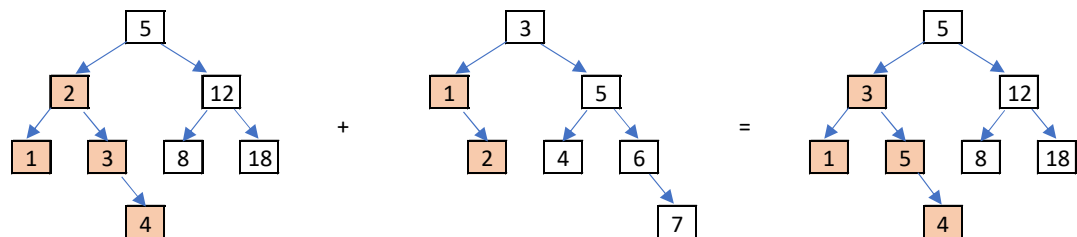


Primer ancestro común: 2

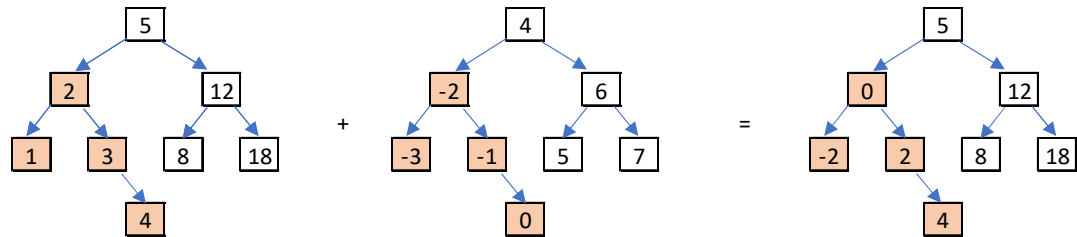
3. Si se encuentra el *primer ancestro común*, el sistema genera otro árbol binario (no necesariamente de búsqueda) llamado *arbol_sistema* y procede a “sumar parcialmente” el *arbol_token* con el *arbol_sistema*. La suma es parcial ya que se realiza sólo al subárbol que tiene como raíz el *primer ancestro común* encontrado. Si la “suma parcial” de ambos árboles genera un árbol binario de búsqueda, el sistema le permite el acceso al usuario.

NOTA: La “suma parcial” de los árboles se hace nodo a nodo según el nivel en el que se encuentra. El árbol resultante se debería manejar como un AB y verificar si es un ABB (Verdadero) o no (Falso).

FALSO



VERDADERO



Se le solicita:

- Establecer la función principal, las estructuras de datos y demás funciones que permitan manejar lo descrito sobre la simulación de verificación del segundo factor. Como resultado final, el usuario debe de recibir un mensaje de “Acceso otorgado” o “Acceso denegado” según sea el caso. (1 punto)
- Elaborar la función `busca_primer_ancestro_comun()` que reciba como parámetros la raíz del ABB `árbol_token` y dos números enteros, que, usando recursividad, devuelva el primer ancestro común o NULL en caso contrario y considerando lo descrito en el punto 2. (4 puntos)
- Elaborar la función `suma_parcial_arboles()` que permita cumplir con el paso 3. La función recibe como parámetros dos árboles binarios y el resultado de la función realizada en b), devuelve verdadero (1) si la “suma parcial” de los árboles binarios resulta un árbol binario de búsqueda, o falso (0) en el caso contrario. (5 puntos)

Pregunta 2 (10 puntos)

Una organización humanitaria acaba de recibir un robot despachador para preparar los envíos de ayuda a los refugiados del medio oriente. El almacén (**nxm**) donde se encuentran los paquetes con la ayuda está desordenado y no tienen una lógica determinada, por tal motivo hay paquetes que se encuentran en medio de otros paquetes, además se sabe que cada paquete tiene un peso determinado.

En cuanto al robot, solo puede avanzar hacia arriba, derecha, abajo e izquierda (sentido horario) por tal motivo se necesita que desarrolle un algoritmo que complemente sus funcionalidades mecánicas con el fin de preparar los envíos para los refugiados, así como un marcado adecuado de las rutas que utilizará dentro del almacén. Se sabe que el robot tiene la capacidad seleccionar uno o más paquetes del almacén llevándolos consigo, siempre y cuando no sobrepasen el peso de la carga solicitada **P**. Se sabe que el robot puede avanzar por el almacén empleando las ubicaciones donde no hay paquetes o aquellas donde el paquete se ha seleccionado y lo lleva consigo (ubicaciones liberadas). Si durante el proceso de selección de paquetes se sobrepasa la carga solicitada o no se puede llegar al peso solicitado, el robot debe colocar los paquetes removidos en el mismo lugar donde se ubicaban inicialmente. Para el inicio de las operaciones se ha considerado siempre emplear la posición (0,0). A continuación, veamos el siguiente ejemplo:

Si recibe como datos de entrada un almacén con $n=9$, $m=5$ y una carga solicitada de $P=29$, considerando los paquetes según su peso distribuidos de la siguiente forma:

	5		5	4
	10		7	2
	7		15	10
8		6	8	
12		10	15	
		20	5	
8	8			

Los números representan el peso de cada paquete

Una solución iniciando del punto (0,0) sería la siguiente:

	5		5	4
			7	2
		6		

1	2	3	4	5
		8	7	6
		9		
		10	11	12
	24	25		13
	23			14
	22			15
	21	20	19	16
			18	17

La solución debe mostrar la distribución de los paquetes seleccionados para llegar el peso P , de la misma forma se debe mostrar la ruta que ha seguido el robot para seleccionar dichos paquetes mostrando el número de pasos que ha dado dentro del almacén. **Recuerde que no puede pasar sobre un paquete sin seleccionarlo.**

Se le solicita implementar lo siguiente:

- Desarrolle una función que utilice backtracking para mostrar (impresión similar al ejemplo) una ruta desde el punto inicial (0,0) a los paquetes a seleccionar, además muestre la ubicación de los paquetes seleccionados que suman el peso P solicitado. Si en caso no es posible completar el peso solicitado debe mostrar un mensaje indicando que no hay solución posible. Son datos de entrada n , m , P y la distribución de los paquetes dentro del almacén (6.0 puntos).

Al revisar las rutas generadas los encargados se dan cuenta que el robot da demasiados pasos para ubicar los paquetes a seleccionar, por tal motivo deciden colocarle un nuevo parámetro para la cantidad de pasos máximos que debe dar el robot para encontrar la solución (**MAX_MOV**). Por ejemplo, si **MAX_MOV** es igual a 10, una de las soluciones será la siguiente:

	5		5	
			7	2
				10

1	2	3	4	
			5	6
				7

- b) Desarrolle una segunda función (separada a la anterior) que modificando el algoritmo de la parte a), devuelva **todas** las soluciones considerando la máxima cantidad de pasos que debe dar el robot para encontrar los paquetes (**MAX_MOV**). Recuerde que en ambas preguntas solo se le solicita imprimir los resultados (4.0 puntos).

Nota: **n, m, P, MAX_MOV, las matrices y su contenido son datos de entrada. Debe realizar cada alternativa como una función separada, si realiza una sola solo se evaluará la de menor puntaje.**

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:

David Allasi
Fernando Huamán
Rony Cueva

San Miguel, 04 de noviembre del 2023