

## Tarea 3. Algoritmos y Estructuras de Datos.

Marco Antonio Heredia Velasco

**Fecha de entrega: Miércoles 18 de junio (durante la clase).**

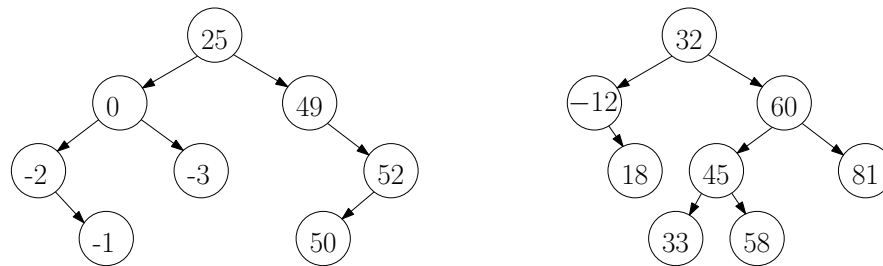
### Instrucciones

Se entregará en **equipo de a lo más 3 personas y a mano**. Por favor, escriban por ambos lados de la hoja.

Responde las preguntas de la siguiente sección. Toda respuesta debe estar plenamente justificada.

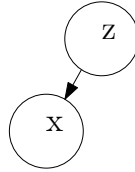
### Preguntas

1. Dada la implementación de Listas Ligadas de enteros en C (vista en clase y enviada a su correo), escribe el código de una nueva función *busca*, que reciba un apuntador a una lista *lst* y un entero *num*, y que entregue el índice de la primera aparición de *num* en *lst* o entregue -1 si *num* no es elemento de *lst*.
2. Por cada uno de los siguientes árboles binarios, escribe el orden en que se imprimen los valores de los nodos, si se recorren desde la raíz en: pre-orden, pos-orden y en-orden.



3. Por cada árbol binario de la pregunta 2, diga si es o no *de búsqueda*, y en caso de serlo, indique el camino que se recorre para determinar si el número 59 pertenece a él.

4. Si  $\perp$  denota un árbol binario vacío, entonces la expresión  $\text{ABin}(\text{ABin}(\perp, x, \perp), z, \perp)$  representa el siguiente árbol binario:



Escriba las expresiones que representan a los árboles binarios de la pregunta 2.

5. Considere un lenguaje de programación que no tiene implementado arreglos, pero sí pilas con sus respectivas operaciones *pop*, *push*, *tope* y *esVacía*. Explique cómo podría implementarse un arreglo unidimensional, con todas sus operaciones, utilizando dos pilas.
6. Supón que tienes dos implementaciones para *Listas*: una simplemente ligada y otra doblemente ligada. En ambas implementaciones tienes apuntadores al nodo inicial y al nodo final. Si tienes una lista de  $n$  elementos, y quieres borrar el último elemento de la lista, ¿cuál de las dos implementaciones efectuaría más operaciones?, ¿de qué orden sería esta operación en cada implementación, en términos de  $n$ ? (Justifique)
7. **(Puntos extra)** Supón que nos dan una pila con  $n$  enteros que queremos ordenar, obteniendo una pila con dichos números ordenados (el elemento más pequeño hasta arriba de la pila). Sólo podemos utilizar las siguientes operaciones:
- *Pop – Push*( $S_1, S_2$ ): Quita el elemento en el tope de la pila  $S_1$  y lo inserta como tope de la pila  $S_2$ .
  - *Compara*( $S_1, S_2$ ): Compara los topes de ambas pilas (regresa 1 si el tope de  $S_1$  es mayor, -1 si el tope de  $S_2$  es mayor y 0 si son iguales).
- a) Da un algoritmo de tiempo  $O(n^2)$  que resuelva el problema utilizando 3 pilas.
- b) Da una configuración inicial de  $n$  números en una pila que no pueda ordenarse utilizando sólo 2 pilas.