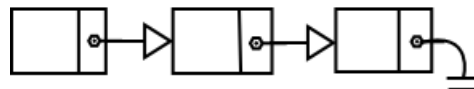


Auxiliar 5 - Listas Enlazadas y Árboles

Profesores: Nelson Baloian
Jeremy Barbay
Patricio Poblete

Auxiliares: Gabriel Flores, Sven Reisenegger
Juglar Díaz, Gabriel Norambuena
Cristóbal Muñoz

P1. Lista enlazada



Lista enlazada

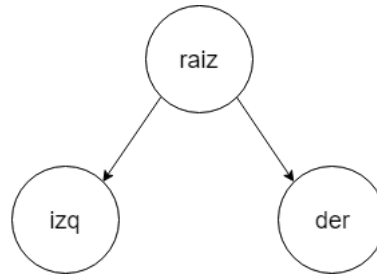
- (a) Defina una clase *ListaEnlazada* que represente una lista enlazada de números enteros. Su constructor debe recibir un `int`, guardarlo en un nuevo nodo y dejar el enlace como `null`. Tome en cuenta que la Lista puede ser circular.
- (b) Escriba un método `public int largo()` que retorne el largo de la lista (es decir, cuantos elementos contiene).
- (c) Defina el método `public int obtener(int i)` que retorne el *i*-ésimo elemento de la lista, si no existe retorne -1.
- (d) Cree el método `public int existe(int num)` que retorne la posición del elemento en la lista y si no está retorne -1.
- (e) Cree el método `public ListaEnlazada eliminar(int i)` que elimine el *i*-ésimo elemento de la lista y retorne el siguiente nodo al eliminado.

P2. Josephus

A Sven le encanta jugar con Listas Enlazadas, un día se aburrió de las Listas Enlazadas normales así que decidió unir el final con el inicio de su Lista Enlazada favorita, quedando una Lista de forma circular.

Sven muy emocionado fue a contarle a su amigo Cristóbal sobre su nueva Lista Circular. Cristóbal quedó muy asombrado por el ingenio de Sven, y le propuso jugar el siguiente juego: Cristóbal le diría a Sven un entero *k*, luego Sven tendría que ir al *k*-ésimo elemento de su Lista Circular y eliminarlo. Luego partiendo del siguiente al que eliminó volver a eliminar el *k*-ésimo, hasta que solo le quede 1 elemento en la Lista. ¿Puedes ayudar a Sven y decir cuál es el último elemento que queda en la Lista Circular? Programe un método `public static int Josephus(ListaCircular L, int k)` que retorne el último elemento de acuerdo al juego dado.

P3. Árboles binarios



Los arboles binarios son una estructura de datos recursiva que se compone de 3 partes, una raíz, un hijo derecho y un hijo izquierdo. Cada hijo es a su vez un árbol binario, por lo que la estructura puede crecer infinitamente. El “caso base” es un árbol vacío representado por null.

- Implemente la clase *ArbolBinario* que almacena un valor `int` en su raíz y recursivamente a otros arboles en sus hijos. Cree 2 constructores
 - `ArbolBinario(int raiz)`
 - `ArbolBinario(int raiz, ArbolBinario izq, ArbolBinario der)`

utilizando la estructura anterior implemente soluciones a los siguientes problemas:

- Obtener el número de nodos.
- Obtener el máximo valor.
- Obtener la suma de todos los valores.
- Obtener el número de “pisos” que tiene.
- Un Árbol Binario de Búsqueda (ABB) es un árbol binario que cumple la siguiente propiedad: todos los nodos a la izquierda de la raíz son menores a la raíz y todos los nodos a la derecha son mayores. Implemente un método que chequee que un árbol binario sea un ABB.