CC30A Algoritmos y Estructuras de Datos
Control 2

Prof. Patricio Poblete, Benjamin Bustos.
Fecha: 18 de mayo de 2001.
Tiempo: 2 horas.

Pregunta 1

Una *deque* es un tipo de dato abstracto que permite almacenar una lista de elementos y realizar operaciones de inserción y eliminación. La definicion de la clase Deque es la siguiente:

```
class Deque {
  // Atributos de la clase
  ...

  // Metodos

  public Deque(){
    // Constructor
  }

  public void push(Object x) {
    // Inserta el elemento x al principio del deque
  }

  public Object pop() {
    // Saca el elemento que se encuentre al principio del deque y lo retorna
  }

  public void inject(Object x) {
    // Inserta el elemento x al final del deque
  }

  public Object eject() {
    // Saca el elemento que se encuentre al final del deque y lo retorna
  }

  public boolean estaVacio(){
    // Retorna true si el deque esta vacio
  }
}
```

Implemente todos los métodos de la clase Deque, definiendo los atributos que sean necesarios, de manera que todas las operaciones tomen tiempo $O(1)$. Utilice un arreglo circular para almacenar los elementos.

Pregunta 2

Suponga que la implementación de los nodos de un árbol de búsqueda binaria (ABB), donde los elementos son números enteros, es la siguiente:

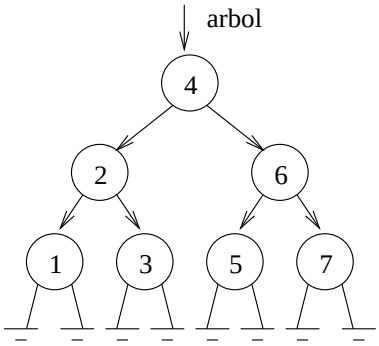
```
class Nodo {
  int elemento;
  Nodo izq;
  Nodo der;
}
```

Implemente el método:

```
Nodo buscarPrevio(Nodo raiz, int x)
```

que recibe por parámetro la raíz de un ABB y un entero x, y retorna una referencia al nodo del ABB cuyo elemento posea el mayor valor que sea menor a x. El método debe devolver *null* en caso que no exista un nodo en el ABB cuyo elemento valga x o cuando x sea el menor de todos elementos en el ABB, puesto que en ese caso no existe un elemento previo.

Para el siguiente ejemplo



- buscarPrevio(arbol, 4) retorna una referencia al nodo 3.
- buscarPrevio(arbol, 5) retorna una referencia al nodo 4.
- buscarPrevio(arbol, 1) retorna null, puesto que el nodo 1 no tiene previo.
- buscarPrevio(arbol, 10) retorna null, puesto que no hay ningún nodo cuyo elemento valga 10.

Indicaciones: Implemente una solución iterativa. Analice cual es la respuesta correcta en distintos casos (cuando x es una hoja, cuando x tiene algun hijo, etc.), y deduzca cuál es la información relevante que debe almacenar en cada paso iterativo.

Pregunta 3

Para un árbol binario, se define el *balance* de un nodo interno como el valor absoluto de la diferencia de alturas de sus dos subárboles hijos. (Nótese que con esta definición, un árbol es AVL cuando todos sus nodos internos tienen balance 0 ó 1.)

Este problema se refiere al *balance promedio* de un árbol, esto es, la suma del balance de todos sus nodos internos dividido por el número de nodos internos.

1. Encuentre una familia infinita de árboles AVL que tengan balance promedio igual a cero.
2. Encuentre una familia infinita de árboles AVL que tengan balance promedio igual a uno.
3. Escriba un método en Java que reciba como parámetro una referencia a la raíz de un árbol binario y que retorne el balance promedio del árbol.