

EjercicioMontículo1

Estructuras de Datos

Tema 4: árboles

1º Grado en Ingeniería de la Computación

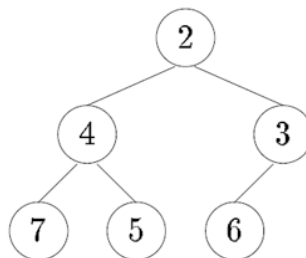
© Profesor Dr. Carlos Grima Izquierdo (www.carlosgrima.com)

URJC (www.urjc.es)

Programa en C++ una clase “Montículo” (cuyos nodos guarden un int) con las siguientes consideraciones:

- La representación interna del árbol ya no es un conjunto de nodos enlazados entre ellos (como en EjercicioGenealógico), sino que ahora los elementos se guardan en un vector contiguo en memoria. Reutiliza la clase ListaContigua (EjercicioListaContigua2). Por lo tanto, la clase Montículo tendrá como atributo interno un objeto de tipo ListaContigua.
- Cada elemento contendrá, en este ejercicio, simplemente su clave de búsqueda: un int.
- Recuerda que los elementos del montículo se numeran desde el 1 (la raíz) hasta el n (la última hoja). El elemento 1 estará en la posición 0 de la lista contigua, y así sucesivamente hasta el elemento n, que estará en la posición n-1.
- Rehaz los dos métodos para imprimir del EjercicioGenealógico para adaptarlo a la nueva representación interna del árbol. Nota que, metiendo estos métodos (que son de interfaz) en la clase Montículo (que es de modelo), estamos violando el principio de separación entre interfaz y modelo (lo hacemos por simplicidad).
- Programa un método privado recursivo que compruebe si un subárbol que empieza en una raíz (cuya posición se pasa como argumento) es un montículo o no (devolverá un booleano). Para ello, recuerda cuál es la definición recursiva de un montículo. Para programarlo, puedes usar la estrategia contraria: en vez de comprobar que cumple todas las características y devolver verdadero en caso afirmativo, puedes devolver falso en el caso de que no cumpla alguna de las características (elige la estrategia que prefieras entre las dos).
- Rehaz el método “insertar” del EjercicioGenealógico para que ahora simplemente pida la clave del nuevo elemento a insertar (por lo tanto ahora sólo tiene un parámetro), metiéndolo en su lugar correcto dentro del árbol. La última línea del método tiene que ser un "assert" para comprobar que la postcondición se cumple (es decir, con el método privado recursivo comprobaremos si, al final del método, lo que ha quedado es un montículo).

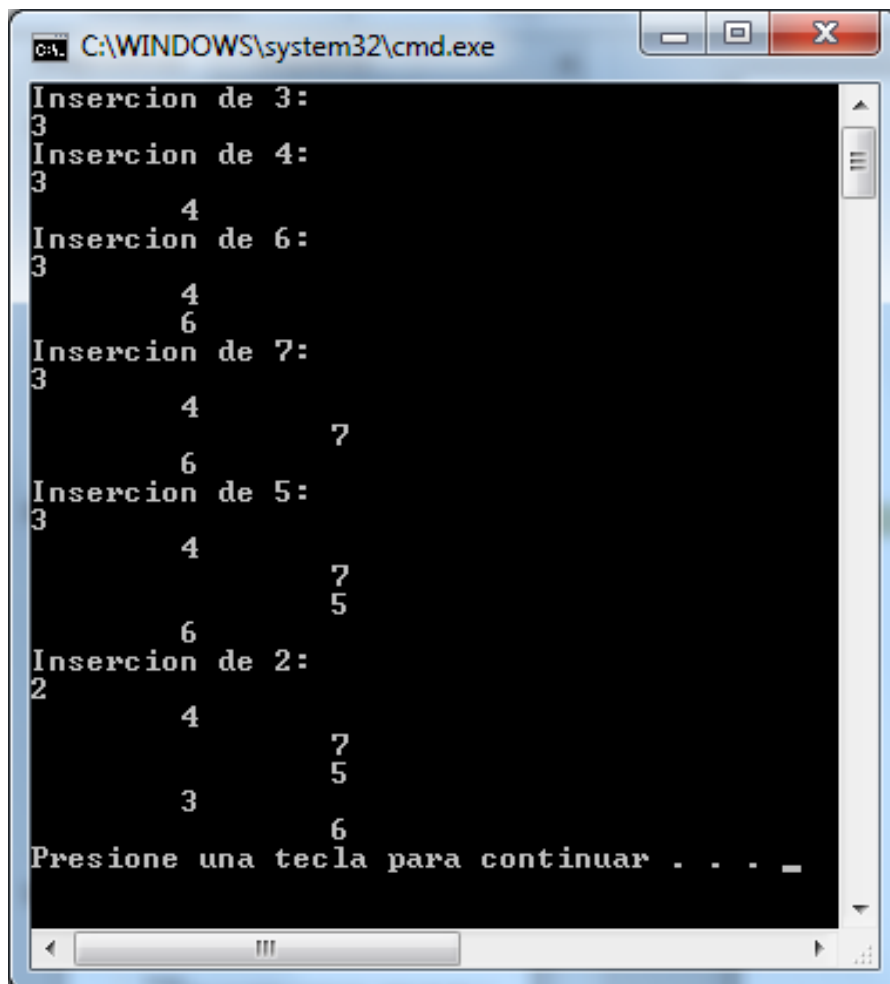
Para probar lo que has hecho, crea un main que forme el montículo siguiente:



Para ello:

1. Meterá el elemento 3 en el montículo (inicialmente vacío)
2. Imprimirá el montículo
3. Meterá el elemento 4 en el montículo
4. Imprimirá el montículo
5. Meterá el elemento 6 en el montículo
6. Imprimirá el montículo
7. Meterá el elemento 7 en el montículo
8. Imprimirá el montículo
9. Meterá el elemento 5 en el montículo
10. Imprimirá el montículo
11. Meterá el elemento 2 en el montículo
12. Imprimirá el montículo.

Todas estas inserciones e impresiones alternadas se programarán con un bucle que recorra el array estático de C++ (3, 4, 6, 7, 5, 2), el cual se ha creado en una sola línea al inicio del main (y así recuerdas y practicas cómo se hace).



```
C:\WINDOWS\system32\cmd.exe
Insercion de 3:
3
Insercion de 4:
3
  4
Insercion de 6:
3
  4
  6
Insercion de 7:
3
  4
    7
  6
Insercion de 5:
3
  4
    7
  6
  5
Insercion de 2:
2
  4
    7
  5
  3
    6
Presione una tecla para continuar . . . _
```