



Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
Semestre I - 2020
Curso CI-0113 - Programación II
Profesor: Edgar Casasola Murillo
Fecha: 14 de julio 2020
Hora: 10 a.m. Duración: 6 horas

EXAMEN II

Para la resolución de este examen usted ya cuenta con la implementación base de un Arbol Roji-Negro (El árbol está implementado dentro del un único archivo fuente llamado ArbolRN.h). Esto se hizo así para facilitarle la programación de su examen. Se le entrega también una aplicación de consulta llamada main.cpp. No es necesario que separe en ningún caso la implementación .cpp de la especificación .h.

El código y el programa main.cpp compilan y se pueden ejecutar en su estado actual desde consola con las instrucción: `g++ main.cpp // para compilar` y `./a.out` o `a.exe` para ejecutar.

El programa mostrará en consola la salida de un árbol binario correspondiente a la inserción de los valores en el Cuadro adjunto. El árbol aparece con una rotación de 90 grados. Los nodos negros tienen una N y los Rojos una R. Las hojas con datos tienen una llave y un valor encerrados en paréntesis cuadrados: [llave : valor], por simplicidad se utilizó una llave tipo int y un valor tipo int con el mismo valor que la llave para fácil visualización. El árbol es un Arbol binario ordenado (menores o iguales a la izquierda y mayores a la derecha). Cada nodo contiene un vector con dos punteros a Nodo para apuntar a sus hijos, IZQ es el índice del hijo izquierdo y DER del derecho). El nodo además cuenta con una llave, un valor, y un char para representar el "color" que puede ser NEGRO o BLANCO. Una hoja es un nodo negro que no tiene hijos. Los nodos intermedios guardan solo llaves y el valor no se utiliza ni inicializan en esos nodos que no son hojas.

SALIDA DEL PROGRAMA	ÁRBOL QUE REPRESENTA
<pre> N [6:6] N 5 N [5:5] R 4 N [4:4] R 3 N [3:3] N 2 N [2:2] N 1 N [1:1] </pre>	

El examen se compone de tres secciones:

Pregunta 1. Árbol equilibrado o Rojo-Negro

- a) (15%) Comprensión del algoritmo de inserción en un Árbol Rojo-Negro
Muestre paso a paso, utilizando dibujos, la secuencia de figuras con las transiciones (debe poner una flecha con la abreviatura de la operación que transforma cada figura en la siguiente) donde queden claros los pasos para construir un Arbol Rojo Negro equilibrado al ir insertando los siguientes 15 valores, en el orden indicado de izquierda a derecha (primero el 9, luego el 1 y así sucesivamente hasta llegar al 2). Se tomará en cuenta cada inserción :

9	1	5	4	8	12	20	3	14	16	10	11	7	6	2
---	---	---	---	---	----	----	---	----	----	----	----	---	---	---

- b) (20 %) Usted debe implementar el método necesario para que el árbol del main.cpp se imprima en forma equilibrada. **Solo debe rellenar EL método NECESARIO (SOLO UNO).** Parte de la evaluación es que usted **determine “cuál” es el método que debe completar** para que se elimine el desequilibrio que se formó en el árbol que aparece en consola. Invoque además el recoloreo respectivo.
Una vez implementado asegúrese que el main.cpp suministrado compile y ejecute correctamente.

Pregunta 2. Clases Parametrizadas o Plantillas.

- a) (20 %) Modifique el ArbolRN.h que se le proporciona para **crear una plantilla de Árbol Rojo Negro emplantillado** y que permite crear Árboles para cualquier combinación de **Llave o Key (K)** con cualquier **valor (V)**. Construya un **main.cpp** utilizando su plantilla donde declare una instancia de clase emplantillada usando su plantilla de clase y agregue los siguientes pares de datos asociando los números del 1 al 7 con su nombre de la siguiente forma:

[1 : “UNO”] [2 : “DOS”] [3 : “TRES”] [4 : “CUATRO”] [5 : “CINCO”] [6 : “SEIS”] [7 : “SIETE”]
--

- b) (15 %) Usando la STL construya una aplicación de consola mainSTL.cpp que utilice un map agregue estos mismos valores y los imprima usando un iterador.

Pregunta 3. Sobrecarga, Herencia y Polimorfismo.

- a) (30 %) Se le ha pedido crear una versión del Arbol Rojo Negro que permita asociar números enteros con Valores pertenecientes a instancias de dos clases diferentes, a saber: instancias de la clase **Entero** e instancias de la clase **Letra**. Se pretende que el programa **mainPolimórfico.cpp que se adjunta funcione con su solución**. Diseñe e implemente un grupo de clases utilizando polimorfismo y sobrecarga de

operadores con el fin de que el mainPolimorfico.cpp funcione correctamente y construya un Arbol equilibrado. Modifique el ArbolRN.h para construir un Arbol Rojo-Negro Polimórfico que asocie números enteros simples (tipo int) con estos Valores creados por usted. programe SOLO LO QUE NECESITA para que el main funcione, permita insertar esos pares de valores y despliegue el árbol equilibrado. No programa métodos innecesarios.

EJEMPLO DE CÓDIGO QUE DEBE FUNCIONAR CON SU ÁRBOL ROJO NEGRO POLIMÓRFICO

```
ArbolRNP arbol;
Entero uno(1);
Entero dos(2);
Entero tres(3);
Letra a('A');
Letra b('B');
Letra c('C');

arbol.insertar(1 , uno);
arbol.insertar(2 , a);
arbol.insertar(3 , tres);
arbol.insertar(4 , b);
arbol.insertar(5 , dos);
arbol.insertar(6 , c);

cout << arbol;
```

FORMA DE ENTREGA:

Debe subir su solución al enlace respectivo en mediacionvirtual.ucr.ac.cr antes de las 6:00 p.m. del día Martes 14 de Julio del 2020.

Su solución puede consistir de:

1. Si hoy no cuenta con computador para la solución del examen:
Carpeta con Fotos o Documento en formato .pdf mostrando en forma legible y clara su solución a puño y letra con el código fuente y la explicación de su solución.
2. Carpeta con subcarpetas (una para cada respuesta a las preguntas del examen).
Debe incluir todo el código .h y .cpps incluido el main asociado a la solución de cada pregunta..
Puede replicar código en las carpetas con tal de que sea compilable sin problema.
Incluya un archivo de texto llamado README.md donde explique como compilar, ejecutar cada solución y posibles problemas encontrados en caso de que no le compile.