

Laboratorio 04

Tema: Arbol Binario de Búsqueda

Profesor	Escuela	Asignatura
Prof. Ric hart Escob e d o Quisp e rescob edo q@unsa.edu.p e	Escuela Profesional de Ingenier ´ ıa de Sistemas	Estructura de Datos y Algoritmos Semestre: I I I C ´ odigo: 1702124

Lab oratorio	T ema	Duraci ´ on
04	Arb ol Binario de B ´ usqueda	02 horas

Semestre acad ´ emico	F ec ha de inicio	F ec ha de en trega
2023 - A	12 Junio 2023	19 Junio 2023

1. Comp etencias del curso

- Espec ´ ıfica: C.m. Construy e r e sp onsablemen te algoritmos sobre estructuras de datos, siguiendo un pro ceso adecuado llev ando a cab o las pruebas a justadas a los recursos d isp onibles.

2. Resultado del estudian te

- RE. 2 La c ap ac id ad de aplicar dise ˜ no de ingenier ´ ıa para pro ducir soluciones a problemas y dise ˜ nar sistemas, comp onen tes o pro c esos para satisfacer necesidades e sp ec ´ ıficas den tro de consideracio- nes realistas e n los asp ectos d e salud p ´ ublica, se gu ridad y bienestar; factores globales, culturales, so ciales, econ ´ omicos y am bien tales .

3. Equip os, materiales y temas

- Sistema Op erativ o (GNU/Linux de preferencia).
- GNU Vim .
- JDK ≥ 1.8 .
- Git .
- Cuen ta en GitHub con el correo institucional.

4. Directorio de traba jo

- Cree su directorio de traba jo.
- Luego, dir ´ ıjase a este directorio, para clonar su rep ositorio y con tin uar sus practicas.

Listing 1: Creando directorio de trabajo

```
$ mkdir -p $ HOME/rescobedoq/
```

Listing 2: Dirijerte a los directorios de trabajo

```
$ cd $ HOME/rescobedoq/
```

Listing 3: Clonando repositorio GitHub

```
$ git clone [URL_DE_SU_GITHUB_PRIVADO]
```

Listing 4: Creando directorio para laboratorio

```
$ mkdir -p $ HOME/rescobedoq/eda-lab-23a/lab04/exercises/
```

- Siempre evalúe utilizar el archivo .gitignore para no considerar algunos archivos innecesarios sobre todo para el repositorio GitHub.
- Pueden haber varios de estos archivos y estar ubicados estratégicamente; por ejemplo sólo para un laboratorio en particular.

Listing 5: Creando .gitignore

```
$ vim $ HOME/rescobedoq/eda-lab-23a/lab04/.gitignore
```

Listing 6: Ejemplo de .gitignore

```
*.class
```

5. Marco teórico

5.1. Árbol Binario

- Los árboles binarios son un tipo de árbol donde cada nodo tiene un máximo de dos grados. Eso significa que cada nodo puede tener como máximo 2 nodos secundarios. Los árboles binarios son una

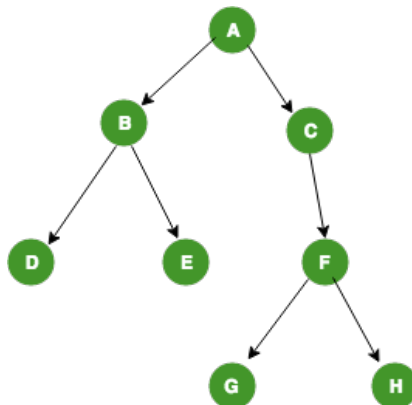


Figura 1: Ejemplo de un árbol binario

5.2. Árbol Binario de Búsqueda (BST)

- El árbol de búsqueda binaria es un árbol binario con la siguiente propiedad: Todos los nodos del subárbol izquierdo de un nodo x son menores o iguales que x y todos los nodos del

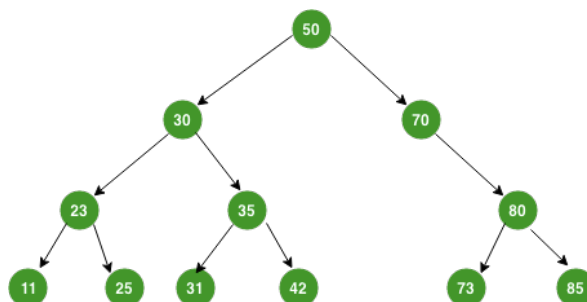


Figura 2: Ejemplo de un árbol binario de búsqueda

5.3. Operaciones en BST

- Buscar()
- Mínimo()
- Máximo()
- Predecesor()
- Sucesor()

- Insertar()
- Borrar()

5.4. No do Genérico

- Ejemplo de No do usando clases genéricas:

Listing 7: No de.java

```
1  class Node<T> {  
2  
3      T data;  
4      Node<T> left;  
5      Node<T> right;  
6  
7 }
```

5.5. BST Genérico

- Ejemplo de BST usando clases genéricas:

Listing 8: BST.java

```
1  class BST<T extends Comparable<T>> {  
2  
3 }
```

6. Tarea

- Elab ore un informe implemen tand o los Binari os de B ´ usqueda con to da la lista de op eraciones search(), getMin(), getMax(), parent(), son(), insert(), remove() .
- INPUT: Una s ´ ola palabra en ma y ´ usculas.
- OUTPUT: Se deb e con truir el ´ arb ol binario de b ´ usqueda considerando el v alor decimal de su c ´ odigo ascii.
- Luego, prueb e to das s u s op eraciones implemen tadas.
- Estudie la libr e ´ r ´ ia Graph Stream para obtener una salida gr ´ afi ca de su impl e men taci ´ on.
- Utilice

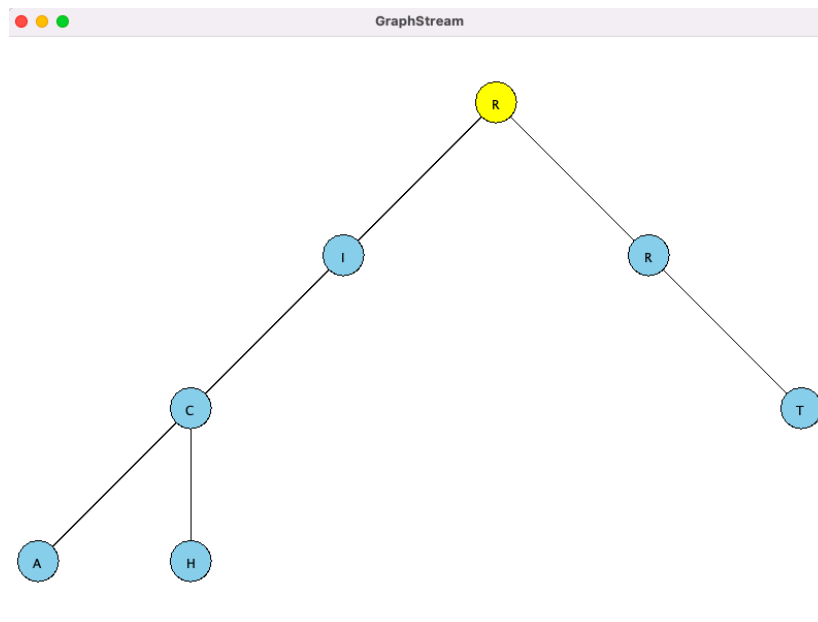


Figura 3: Ejemplo de un ´ arb ol binario de b ´ usqueda generado

7. Pregunta

- ¿Explique como es el algoritmo que implemen t ´ o para obtener el ´ arb ol binar io de b ´ usqueda con la librer ´ ia Grap h Stream? Recuerde que pueden hab er op eraciones sobre el BST.

8. En tregables

- El informe debe tener un enlace al directorio específico del laboratorio en su repositorio GitHub privado donde esté todo el código fuente y otros que sean necesarios. Evitar la presencia de archivos: binarios, objetos, archivos temporales, caché, librerías, en tornos virtuales. Si hay configuraciones particulares puede incluir archivos de especificación como: leeme.txt, o importante.txt.
- No olvide que el profesor debe ser siempre colaborador a su repositorio (Usuario del profesor @rescobedoq).
- Para ser considerado con la calificación de máxima nota, el informe debe estar elaborado en LaTeX .
- Usted debe describir sólo los commits más importantes que marcaron hitos en su trabajo, adjuntando capturas de pantalla, del commit, del código fuente, de sus ejecuciones y pruebas.
- En el informe siempre se debe explicar las imágenes (código fuente, capturas de pantalla, commits, ejecuciones, pruebas, etc.) con descripciones puntuales pero precisas.

9. Rúbricas

9.1. Rúbrica para el tregable Informe

TTabla 1: Rúbrica para tipo de Informe

Informe		Cumple	No cumple
Latex	El informe está en formato PDF desde Latex, con un 20 formato limpio (buena presentación) y fácil de leer.		0
Markdown	El informe está en formato PDF desde Markdown 17 README.md, con un formato limpio (buena presentación) y fácil de leer.		0
MS Word	El informe está en formato PDF desde plantilla MS 15 Word, con un formato limpio (buena presentación) y fácil de leer.		0
Observaciones	Por cada observación se le descontará a puntos. -	-	-

9.2. Rúbrica para el contenido del Informe y demostración

- El alumno deberá marcar o dejar en blanco en las celdas de la columna Checlist, de acuerdo a si cumplió o no con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación siempre será a sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe auto calificarse en la columna Estudian te de acuerdo a la tabla de calificación de niveles de desempeño:

Tabla 2: Niveles de desempeño

Nivel				
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

T abla 3: R úbrica par a con tenido del Informe y demostraci ón

Con tenido y demostraci ón		Pun tos	Chec klist	Estudian te	Profesor
1. GitHub	Ha y enlace URL activ o del directorio para el lab oratorio hacia su rep ositorio GitHub con c ódigo fuen te terminado y f ácil de revisar.	2			
2. Commits	Ha y capturas de pantalla de los commits m ás imp ortan tes con sus explicaciones detalladas. (El profesor puede pregun tar para refrendar calificaci ón).	4			
3. C ódigo fuen te	Ha y p orciones de c ódigo fuen te imp ortan tes con n umeraci ón y explicaciones detalladas de sus funciones.	2			
4. Ejecuci ón	Se incluy en ejecuciones/pruebas del c ódigo fuen te e xp licadas gradualmen te.	2			
5. Pregun ta	Se resp onde con completitud a la pr e gu n ta for m ulada en la tarea. (El profesor puede pregun tar para refr e n dar calificaci ón).	2			
6. F ec has	Las fec has de m o dificaci ón del c ódigo fuen te estand en tro de los plazos de fec h a de en trega establecidos.	2			
7. Ortograf ía	El do cumen to no m uestra errores ortogr áficos.	2			
8. Madurez	El Informe m uestra de manera general u na ev oluci ón de la madurez del c ódigo fuen te, ex plicaciones pun tuales p e r o precisas y un aca bado imp ecable. (El profesor puede pregun tar para refrendar calificac i ón).	4			
T otal		20			

10. Referencias

- <https://algorithmtutor.com/Data-Structures/Tree/Binary-Trees/>
- <https://algorithmtutor.com/Data-Structures/Tree/Binary-Search-Trees/>
- https://github.com/rescobedoq/graphstream_examples