

LABORATORIO NO. 8: TABLAS DE SÍMBOLOS ORDENADAS y BALANCEADAS

1 OBJETIVOS

Incluir en el diseño de soluciones el uso de *Maps* o Tablas de símbolos ordenadas y utilizar los árboles binarios de búsqueda balanceados (*RBTs*) para su implementación

- a) Analizar los órdenes de crecimiento y el desempeño de los *RBTs* como índices ordenados para la consulta de datos
- b) Integrar los *RBTs* con las otras estructuras de datos vistas en el curso

2 DESARROLLO

En los grupos previamente definidos sigan los siguientes pasos para el laboratorio de hoy.

2.1 CARGAR EL EJEMPLO ISIS1225-SAMPLETREE

Asegúrense que todos los miembros del equipo tienen acceso al ejemplo, en caso de que no lo hayan hecho para el laboratorio 7. Para ello utilicen el URL <https://github.com/ISIS1225DEVS/ISIS1225-SampleTree> y sigan el proceso acostumbrado: Fork del proyecto a su espacio en GitHub y clone a su máquina local. Si ya tienen el ejemplo, actualícenlo a la nueva versión del ejemplo.

2.2 EJECUTAR EL EJEMPLO

Cuando tenga los datos de prueba, ejecute el programa, desde el archivo `view.py` tal como lo hicieron en el laboratorio 7. Ejecute primero la opción 1 que inicializa las estructuras de datos, luego la opción 2, para cargar la información.

Al cargar la información, verá el número de registros cargados, la menor llave encontrada, la mayor llave encontrada, la altura del árbol (*BST*) y el número de nodos en el árbol (*BST*).

Tome nota del total de elementos en el árbol y la altura del *BST* reportada. Recuerde a que conclusión llegaron en el laboratorio anterior sobre estos valores.

Número de elementos (*BST*): 1177

Altura del árbol (*BST*): 29

2.3 EJECUTAR EL EJEMPLO CON UN RBT

Salgan del programa y ahora editen el archivo `model.py`, específicamente donde se crea el BST.

```
analyzer['dateIndex'] = om.newMap(omaptype='BST',  
    comparefunction=compareDates)
```

Cambien el valor del parámetro **omaptype** y ahora usen un árbol rojo-negro (RBT), como se muestra en el ejemplo.

```
analyzer['dateIndex'] = om.newMap(omaptype='RBT',  
    comparefunction=compareDates)
```

Ahora, ejecuten nuevamente el programa desde el archivo `view.py`. Seleccionen la opción 1 y luego la opción 2.

Nuevamente verán el número de elementos en el árbol y la altura del árbol.

Número de elementos (RBT): 1177

Altura del árbol (RBT): 13

Pregunta 1: ¿Qué diferencia existe entre las alturas de los dos árboles (BST y RBT) ?, ¿por qué pasa esto?

La diferencia que existe entre las alturas de los dos árboles es de 16 niveles. Esto es debido a que el RBT es un árbol balanceado, mientras que el BST no lo es.

2.4 MODIFICAR EL REQUERIMIENTO 1 DEL RETO 3

Utilicen la implementación del requerimiento 1 del reto 3 del laboratorio anterior (Lab 7) en el que utilizaron un BST, para averiguar la altura del árbol. Hagan la prueba con los datos del año 2016.

Pregunta 2: ¿Cuántos elementos tiene el árbol (size)? ¿Qué altura tiene el árbol (height)?

Número de elementos (BST): 344

Altura del árbol (BST): 14

2.5 MODIFICAR EL REQUERIMIENTO 1 DEL RETO 3

Cambien el requerimiento 1 del reto 3, desarrollado en el laboratorio anterior para que ahora utilice un RBT.

Pregunta 3: Qué tan difícil fue hacer el cambio de una estructura de datos por otra? ¿Cuántas líneas de código tuvieron que modificar para hacer el cambio?

La dificultad de hacer el cambio de una estructura de datos por la otra fue minima y solo tuvimos que modificar una parte de una linea de codigo. Cambiar 'BST' por 'RBT'

Ahora prueben con el mismo archivo de datos (año 2016) la nueva estructura de datos (RBT)

Pregunta 4: Cuántos elementos tiene el árbol? ¿Qué altura tiene el árbol? ¿Qué puede concluir sobre las alturas de los árboles cuando se usa un BST o un RBT?

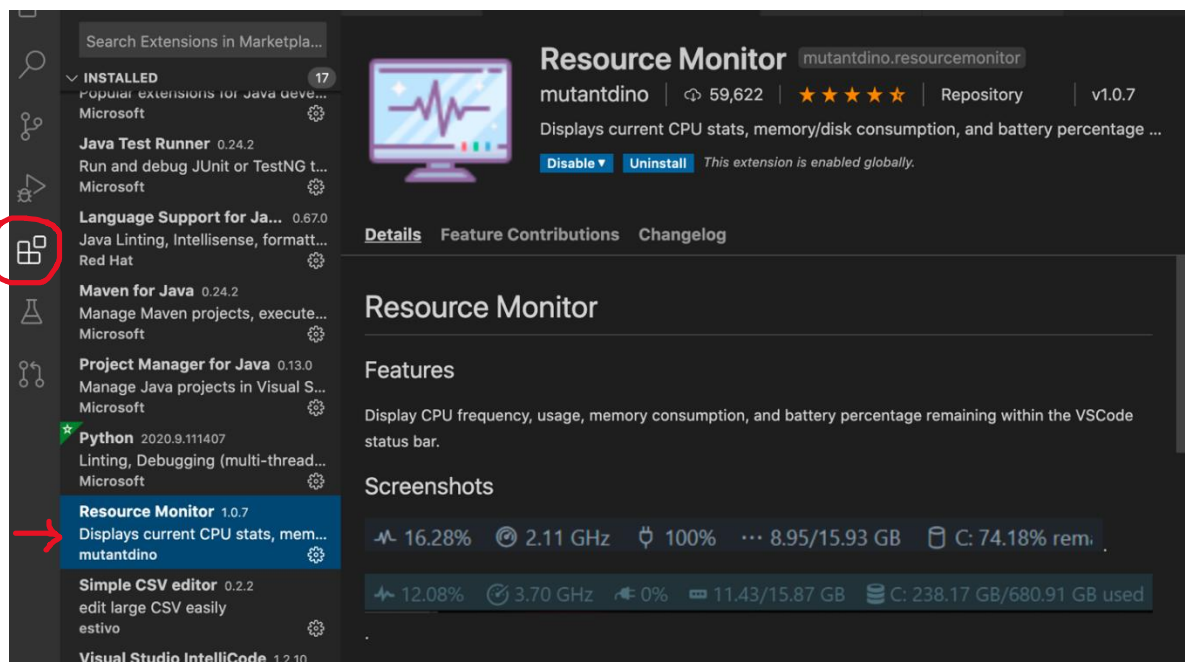
Número de elementos (RBT): 344

Altura del árbol (RBT): 11

Se puede concluir que la altura de un arbol RBT normalmente es menor que la altura de un arbol BST, ya que el RBT es un arbol balanceado, mientras que el BST no lo es.

2.6 UTILIZAR EL CONJUNTO COMPLETO DE DATOS CON EL REQUERIMIENTO 1 DEL RETO 3

Ahora, instalen en su VS Code cualquier plugin de ayuda para la verificación de uso de los recursos disponibles de su computador (procesador, memoria, etc.). Se sugiere la utilización de "Resource Monitor".



En la barra inferior de VSCode verán la información de los recursos de su computador:



Tomen nota de la memoria utilizada que reporta VSCode

Mem Inicial: 3.50 GB

Ejecuten el reto 1 del requerimiento 3 (solo usando un RBT), cargando los datos de un solo año (2016) e inspeccionen la cantidad de memoria utilizada (con el programa aun ejecutándose)

Mem Final: 4.10 GB

La diferencia es la memoria utilizada durante la ejecución de su programa

Mem Utilizada: 0.60 GB

Terminen el programa y ejecuten esta misma medición varias veces (por ejemplo 3) y calculen un promedio.

Ahora corran de nuevo el programa, pero esta vez utilicen todos los datos del proyecto (archivo completo con los 4 años) y ejecuten el mismo requerimiento 1 del reto 3. Inspeccionen nuevamente el uso de la memoria reportada en VSCode.

Pregunta 5: Existe diferencia en el consumo de memoria? ¿Pueden proponer una relación entre el total de datos cargados y la memoria utilizada?

Es claro que existe una diferencia en el consumo de memoria. Esto se debe a que el archivo completo es mucho mas pesado (1.12 GB), mientras que el archivo de un solo año (2016) es menos pesado (49.6 MB).

2.7 COMPARTIR EL PRODUCTO DE LA PRACTICA CON LOS EVALUADORES

El resultado de este laboratorio es la implementación del requerimiento 1 del reto 3. Para entregar exitosamente sus resultados de este laboratorio, por favor recuerde las siguientes indicaciones:

- Crear un Commit en el depósito de su reto 3 con el formato **EDA-2020-20-Lab-08-SEC-<<Número de La sección>>-GRUPO-<<Número del grupo>>**.
- Invitar a los monitores del laboratorio asignados.
- Incluir en el **README** del repositorio los datos completos de los integrantes del grupo (nombre completo, correo Uniandes y código de estudiante).
- Incluir en la carpeta *Docs* un documento en formato PDF que indique lo siguiente:
 - Datos completos de los integrantes del grupo (nombre completo, correo Uniandes y código de estudiante).
 - Las preguntas del laboratorio con sus respectivas respuestas, en un documento PDF llamado *“respuestas-Lab8.pdf”*, marcado con el nombre de

los integrantes del grupo. Dentro del documento la respuesta a las preguntas:

Pregunta 1: Qué diferencia existe entre las alturas de los dos árboles (BST y RBT) por qué pasa esto?

La diferencia que existe entre las alturas de los dos arboles es de 16 niveles. Esto es debido a que el RBT es un arbol balanceado, mientras que el BST no lo es.

Pregunta 2: ¿Cuántos elementos tiene el árbol (size)? ¿Qué altura tiene el árbol (height)?

Número de elementos (BST): 344

Altura del árbol (BST): 14

Pregunta 3: Qué tan difícil fue hacer el cambio de una estructura de datos por otra? ¿Cuántas líneas de código tuvieron que modificar para hacer el cambio?

La dificultad de hacer el cambio de una estructura de datos por la otra fue minima y solo tuvimos que modificar una parte de una linea de codigo. Cambiar 'BST' por 'RBT'

Pregunta 4: Cuántos elementos tiene el árbol? ¿Qué altura tiene el árbol? ¿Qué puede concluir sobre las alturas del árbol cuando se usa un BST y un RBT?

Número de elementos (RBT): 344

Altura del árbol (RBT): 11

Se puede concluir que la altura de un arbol RBT normalmente es menor que la altura de un arbol BST, ya que el RBT es un arbol balanceado, mientras que el BST no lo es.

Pregunta 5: Existe diferencia en el consumo de memoria? ¿Pueden hacer una relación entre el total de datos cargados y la memoria utilizada?

Es claro que existe una diferencia en el consumo de memoria. Esto se debe a que el archivo completo es mucho mas pesado (1.12 GB), mientras que el archivo de un solo año (2016) es menos pesado (49.6 MB). La relacion que existe entre el total de datos cargados y la memoria principal es de $O(n)$.

Recuerden que cualquier documento solicitado durante las actividades debe incluirse en el repositorio GIT y que solo se calificará hasta el último **COMMIT** realizado dentro de la fecha límite del miércoles 14 de octubre de 2020, antes de la media noche (11:59 pm).