

Informe de Laboratorio 07

Tema: Trie

Nota

Estudiante	Escuela	Asignatura
Victor Gonzalo Maldonado Vilca, Armando Steven Cuno Cahuari vmaldonadov@unsa.edu.pe, acunoc@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Estructura de Datos y Algoritmos Semestre: III Código: 1702122

Tarea	Tema	Duración
07	Trie	2 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 04/07/24 – 09:42am	Al 04/07/24 – 23:59pm

1. Introducción

Los árboles Trie son estructuras de datos eficientes para almacenar y recuperar cadenas de caracteres, especialmente útiles para operaciones rápidas de búsqueda y autocompletado. En este documento, se explora la implementación de un árbol Trie en Java, enfocándose en su diseño y operaciones fundamentales.

2. Objetivos

- **Implementación del Árbol Trie:** Desarrollar un árbol Trie eficiente en Java, incluyendo métodos para inserción, búsqueda y eliminación de cadenas.
- **Análisis de Complejidad:** Evaluar la complejidad temporal y espacial de las operaciones del árbol Trie implementado.
- **Aplicaciones Prácticas:** Explorar aplicaciones como autocompletado de texto y gestión de listas de palabras clave usando árboles Trie.

3. Tarea

- Elabore un informe paso a paso de la implementación un Trie para insertar, buscar y reemplazar palabras en un texto.
- Encuentra las primeras 'k' palabras que ocurren con mayor frecuencia en un conjunto dado de cadenas(que se insertaron previamente en el Trie)

4. Entregables

- Informe hecho en Latex.
- URL: Repositorio GitHub.
- Archivos Java.

5. Equipos, materiales y temas utilizados

- Trie
- Git
- notepad++
- Latex
- Java

6. URL de Repositorio Github

- Link: GitHub.
- https://github.com/Victor-Gonzalo-Maldonado-Vilca/EDA_lab07.git

7. Desarrollo del trabajo

8. Conclusiones

8.1. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 1: 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

Tabla 2: Rúbrica para contenido del Informe y demostración

	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
Total		20		20	

9. Referencias

- <https://www.baeldung.com/trie-java>
- <https://www.youtube.com/watch?v=fUpZ05dNZdE>
- <https://www.w3schools.com/java/>
- <https://www.eclipse.org/downloads/packages/release/2022-03/r/eclipse-ide-enterprise-java-and-webtools>
- <https://docs.oracle.com/javase/7/docs/api/java/util/List.html>
- <https://docs.oracle.com/javase/tutorial/java/generics/types.html>