



Estructuras de Datos (2022-1) Proyecto 2: Mapas

Profesor: Alexander Iribarra

Ayudantes: Leonardo Aravena, Diego Gatica, Vicente Lermenda

Objetivos

Los objetivos de este proyecto son:

- Mejorar la programación, compilación y ejecución de programas escritos en lenguaje *C++* u otros.
- Estudiar estructuras de datos basadas en árboles.
- Implementar y analizar estructuras de datos que implementan mapas.

1. Descripción del problema

El ADT Mapa ha sido estudiado en la asignatura y se ha implementado en uno de los laboratorios a través de tablas hash, además de que se han revisado varias aplicaciones en las que se puede utilizar. En este proyecto, se expandirá el número de implementaciones, considerando además de las ya realizadas las implementaciones a través de árboles AVL y vector ordenado.

Este proyecto está enlazado con el Laboratorio 7, del cual se utilizará la mejor implementación de tabla hash, entre las cuales se escogerá la mejor implementación hecha por cada integrante de la pareja.

2. Implementación

1. Se creará una clase abstracta **Map** que funcionará como interfaz para los métodos básicos, es decir:
 - **Insertar un par:** `virtual void insert(const string&, int)=0;`
 - **Eliminar el par clave-valor asociado a una clave:** `virtual void erase(const string&)=0;`

- **Encontrar el valor asociado a una clave k :** `virtual int at(const string&)=0;`
- **Obtener la cantidad de elementos almacenados:** `virtual int size()=0;`
- **Verificar si está vacío:** `virtual bool empty()=0;`

Se deben realizar las siguientes implementaciones:

- a) *MapSV*: Una implementación que utilice un vector que se mantiene ordenado. La operación de búsqueda se debe implementar a través de búsqueda binaria. Se sugiere utilizar las implementaciones de vector¹ y de búsqueda binaria² disponibles en los lenguajes de programación.
- b) *MapHash*: Implementación que utiliza tablas hash. Se debe escoger la mejor variante realizada para el laboratorio 7, y de esta se escogerá la mejor implementación de los integrantes del equipo. En el informe se debe mencionar la decisión y justificarla.
- c) *MapAVL*: Implementación basada en árboles AVL.

Para todas estas implementaciones se recomienda revisar la bibliografía del curso, pues en este material se encuentran pseudocódigos que facilitan la tarea de implementar las estructuras solicitadas.

2. Se debe hacer un análisis teórico para obtener las complejidades de cada método en cada implementación.
3. Se realizará un análisis experimental de las soluciones. Para esto, se recomienda utilizar un generador de cadenas al azar de largo fijo. Se debe medir el tiempo promedio de n operaciones, midiendo separadamente cada método. En el informe debe quedar detallada la metodología de los experimentos.

Observación

Los estudiantes pertenecientes al minor son libres de implementar las soluciones en el lenguaje de programación *C++*, *Java* o *Python*.

3. Evaluación

El proyecto se realizará en parejas. Se debe subir a Canvas lo siguiente:

1. Un informe que:

¹En C++ <https://es.cppreference.com/w/cpp/container/vector>, en Python *list* con todos sus métodos

²En C++ https://es.cppreference.com/w/cpp/algorithm/lower_bound, en Python <https://docs.python.org/3/library/bisect.html>

- a)* Incluya portada, descripción de la tarea, descripción de las soluciones propuestas, detalles de implementación, análisis teórico y análisis experimental.
 - b)* Sea claro y esté bien escrito. Un informe difícil de entender será mal evaluado, aunque todo esté bien implementado. Quien revise el documento debe poder entender su solución solo mirando el informe.
 - c)* Esté en formato pdf.
- 2. Un archivo comprimido con todos los ficheros fuente implementados para solucionar la tarea. El informe debe hacer referencia a ellos y explicar en qué consiste cada uno.

Fecha de entrega: viernes 15 de julio de 2022 11:59PM