



Estructuras de Datos (2021-1)

Laboratorio 6

Profesor: Alexander Iribarra

Ayudantes: Leonardo Aravena, Diego Gatica, Vicente Lermenda

Objetivos

Los objetivos del laboratorio son:

- Mejorar la programación, compilación y ejecución de programas escritos en lenguaje *C++* u otros.
- Implementar y analizar diferentes estructuras de datos y algoritmos de ordenamiento.

Ejercicios

Dado el tipo de dato abstracto **PriorityQueueADT**, el cual contiene los siguientes métodos:

- **Insertar elemento:** `virtual void push(int)=0;`
- **Obtener el elemento mínimo:** `virtual int top()=0;`
- **Eliminar el elemento mínimo:** `virtual void pop()=0;`
- **Obtener la cantidad de elementos almacenados:** `virtual int size()=0;`
- **Verificar si está vacío:** `virtual bool empty()=0;`

1. Implementar la estructura de datos **PriorityQueueUnsorted**, que debe heredar de la clase **PriorityQueueADT** e implementar sus métodos, utilizando un vector desordenado (Se deben crear los archivos `PriorityQueueUnsorted.h` y `PriorityQueueUnsorted.cpp`).

Se puede utilizar la clase *vector*¹ en C++ con sus respectivos métodos. Similarmente, se puede usar *ArrayList* para entregas en Java y *array*² en Python.

¹<https://www.cplusplus.com/reference/vector/vector/>

²<https://docs.python.org/3/library/array.html>

2. Implementar la estructura de datos **PriorityQueueHeap**, que debe heredar de la clase **PriorityQueueADT** e implementar sus métodos, utilizando un min-heap (Se deben crear los ficheros `PriorityQueueHeap.h` y `PriorityQueueHeap.cpp`). La implementación debe ser basada en vectores y se pueden usar las mismas estructuras mencionadas en el punto anterior.
3. Indicar la complejidad de los métodos *push*, *top* y *pop* de ambas implementaciones.
4. Implementar los algoritmos de ordenamiento **Heap Sort** y **Selection Sort** utilizando las estructuras creadas.
5. Realizar un análisis experimental de los algoritmos de ordenamiento, utilizando vectores de distintos tamaños. Escribir sus resultados en una tabla y realizar un gráfico comparativo.

Observación

Los estudiantes pertenecientes al minor son libres de implementar las soluciones en el lenguaje de programación *C++*, *Java* o *Python*, en este caso pueden tomar el código proporcionado como una base para empezar a realizar los ejercicios.

Normas de entrega

Antes del subsiguiente laboratorio, se deben enviar todos los ejercicios resueltos a los ayudantes mediante la plataforma CANVAS.

Se debe entregar un archivo comprimido que contenga:

- Archivo PDF con el nombre completo, número de matrícula, las respuestas a las preguntas que correspondan.
- Todos los ficheros del código fuente.
- **IMPORTANTE:** Los archivos debe llamarse *apellido1_nombre_06.formato*