Juan Fernando Martínez Hidalgo A00358232

Lab 4 – Algorithm's Race

GitHub Repository Link: https://github.com/JuanF2019/basic-algorithm-race

1. Functional requirements

a. Add N elements to different data structures starting with 0 elements.
b. Find N elements on different data structures starting with N random elements.
c. Remove N elements from different data structures starting with N random elements.
d. Take the time it took each one of the add, find or remove processes.
e. Show in screen a general chronometer that measures in real time how much time it takes to end all the add, find or remove processes.
f. Show in screen three chronometers that show the time it has been taken to complete an add, find or remove process in each data structure.
g. Animate two circles so that when one is growing the other is shrinking. It only runs the animation when any of the processes is running.
h. Show in screen the progress of the process in each data structure, using percentage or a progress bar.

2. Nonfunctional requirements

a. Add a long type value to a double linked list using iterative and recursive algorithms.
b. Add a long type value to an ArrayList.
c. Add a long type value to a binary search tree using iterative and recursive algorithms.
d. Remove a long type value from a double linked list using iterative and recursive algorithms.
e. Remove a long type value from an ArrayList using recursive and iterative algorithms.
f. Remove a long type value from a binary search tree using iterative and recursive algorithms.
g. Find a long type value in a double linked list using iterative and recursive algorithms.
h. Find a long type value in an ArrayList using iterative and recursive algorithms.
i. Find a long type value in a binary search tree using iterative and recursive algorithms.
j. Count the time using System.currentTimeMillis(); for using chronometer.
k. Each of the add, remove or find processes will run using three different threads, one for each data structure.
l. Show everything using a GUI implemented with JavaFX.

3. Class Diagram (Some relations are written as attributes in order to get a better visual)