

DATA STRUCTURES

1. Description

The library that you implement must support the following data structures:

pile of minimums, **pile of maximums**, **binary AVL search tree**, **chart without directions and no edge weights**, **hash table**. For each structure, there should be an .h file which should be #include by the program that wants to use the corresponding structure. Also, for each structure there should be at least one .cpp file that implements the corresponding structure.

Your program will not read anything from the keyboard and also will not print anything on the screen. When the program starts, it should read the file commands.txt which contains one command on each line. Your program should execute each command depending on the content and output to output.txt file. The commands it can contain are as follows:

BUILD MINHEAP filename	Build a bunch of minimums by reading the data from the filename file.
BUILD MAXHEAP filename	Build a pile of maxims by reading the data from the filename file.
BUILD AVLTREE filename	Construction Binary Tree AVL Search Reading the data from the filename file
BUILD GRAPH filename	Construct a chart by reading the data from the filename file.
BUILD HASHTABLE filename	Build a hash table by reading the data from the filename file
GETSIZE MINHEAP (similarly for MAXHEAP, AVLTREE, GRAPH, HASHTABLE)	Returns the number of items in the pile of minimums. In the case of the graph, two values are returned, the number of vertices and the number of edges.
FINDMIN MINHEAP	Finding the minimum element from the minimum pile.
FINDMAX MAXHEAP	Finding the maximum element from the maximum pile.
FINDMIN AVLTREE	Find the minimum element from the AVL tree.
SEARCH AVLTREE number	Search for the number item in the AVL tree. If the item exists, the string SUCCESS or FAILURE is returned.
SEARCH HASHTABLE number	Search for Number in the table
	Hash. If the item exists, the string SUCCESS or FAILURE is returned.
COMPUTESHORTESTPATH GRAPH number1 number2	Calculates the cost of the minimum trip from node number1 to node number2.
COMPUTESPANNINGTREE GRAPH	It calculates the cost of the minimum extended tree of the chart and returns it.
FINDCONNECTEDCOMPONENTS GRAPH	Calculates the number of coherent components of the chart and returns that number.

INSERT MAXHEAP number (similarly for MAXHEAP, AVL TREE, HASHTABLE)	Inserts the number number into the pile of minimums.
INSERT GRAPH number1 number 2	Inserts the acne <number1, number2> into the chart if the acne does not exist.
DELETEMIN MINHEAP	Delete a minimum from MINHEAP.
DELETEMAX MAXHEAP	Delete maximum from MAXHEAP.
DELETE AVL TREE number	Delete an item from AVL.
DELETE GRAPH number1 number2	Delete the edge <number1, number2> from the chart if it exists.

Note: do not implement deletion in HASHTABLE.

Each command in file should be executed using appropriate methods from the data structures that you will implement. **The file should be read inside main().** The result of each command should be written to the unique output.txt file along with the execution time. It is noted that the order of the commands in the commands.txt file may be random. However, the file will always be correct, in the sense that it will not contain grammatical errors, and also in order for there to be a command to use a structure, the structure must first be built with the BUILD command.