CMPT 225 SECTION D100

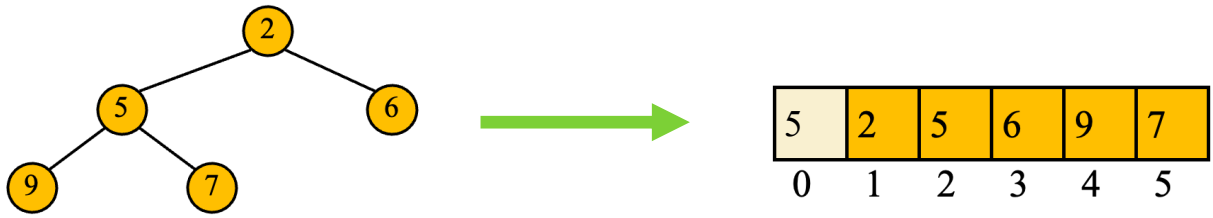
ASSIGNMENT #3: DYNAMIC HEAPS

DUE DATE WITH CANVAS: SUNDAY, JUNE 15, 2025 AT 11:59 PM

PROBLEM AT HAND

For this assignment, you will finish implementing a max-heap that stores int keys. The max-heap will use an array to represent the binary tree structure. Index 0 stores the size (number of values stored). The root element, if present, is at index 1. Child nodes of the node at index i are at $2i$ and $(2i + 1)$.

Example:



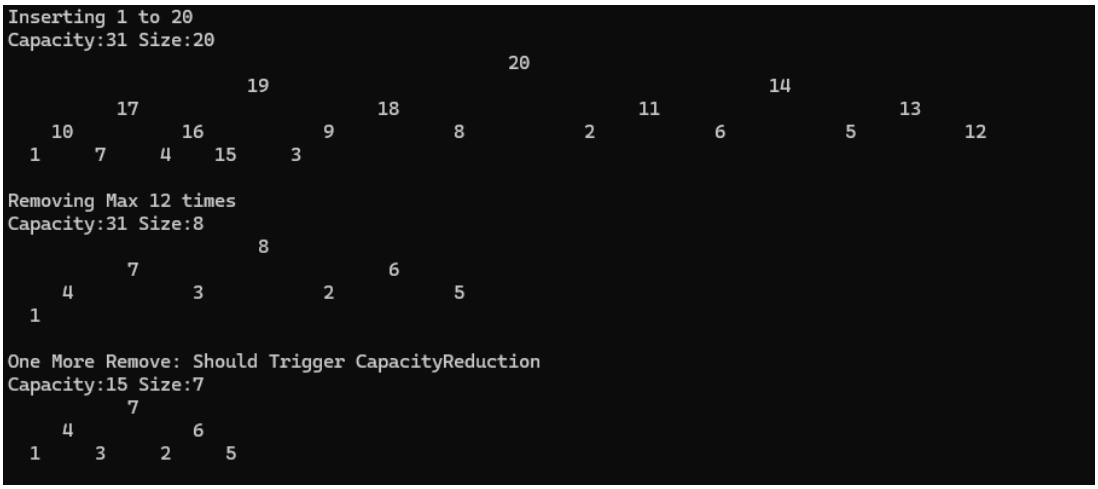
The functions you will need to complete are:

- The Big Three: copy constructor, assignment operator, destructor
- `insert` – adds element to the heap, may invoke `increaseCapacity` if more space is needed
- `increaseCapacity` – private function to double the heap capacity
- `removeMax` – removes the root from the max-heap, may invoke `decreaseCapacity` if less than ¼ of the array contains elements.
- `decreaseCapacity` – private function to shrink the heap capacity in half

The initial array starts with a capacity for 7 values (length 8). If it is filled and insert is called, the increaseCapacity method should be called to increase the capacity to (2*capacity)+1. Meaning the capacity will increase in the order: 7, 15, 31, etc.

When removing the max value, if the size is less than one quarter of the capacity, decrease the capacity the capacity to capacity/2. Meaning the capacity will decrease in the order: 31, 15, 7. (Capacity will never go less than 7)

The main function given tries to draw the max-heap (as best as we can in the console).



ASSIGNMENT STRUCTURE

- The Heap class is defined in: DynamicHeap.h and DynamicHeap.cpp. The implementation of each function must be written in DynamicHeap.cpp. Do not change any of the current public members or private data in the header file, but you can add private helper functions. Helper functions must be defend in DynamicHeap.h and implemented in DynamicHeap.cpp
- You must only edit the `DynamicHeap.cpp`
- To make and run the file for each problem, enter the commands:
  - `make`
- To clean up unneeded object files and executables, enter the command:
  - `make clean` (for windows)
  - `make remove` (for mac/linux)

TO SUBMIT WITH CANVAS AS A SINGLE ZIP FILE:

Remove any executables or object files from your project folder.

Your folder should only contain:

- DynamicHeap.h
- DynamicHeap.cpp
- assign3.cpp
- makefile

Zip this folder and name it as: `firstname_lastname_studentID.zip` and submit this file on Canvas by the due date. Any last submission will result in a 10% penalty per day. No late assignments will be accepted after 48 hours.