Lab 11

100 Points

Due date:
11:59pm, Monday 04/29/2019 for Tuesday labs.
11:59pm, Wednesday 05/01/2019 for Thursday labs.

Purpose:
The purpose of this lab is to implement an experimental profiling on the Min-5 Heap, Max-5 Heap, and Binary Search Tree (BST) data structures.

General Requirements:
For this lab, you are required to implement an experimental profiling on Min-5 heap, Max-5 heap, and BST from Lab 8 and Lab 5. First, you need to make sure your implementations of both heaps and the BST are correct. You need to correct your code before you can compare the data structures' performance. After you verify the correctness of your heaps and BST, you can move on to the performance comparison. In this lab, we will compare the runtime performance for build, deleteMin, and deleteMax operations for Min-5 heap, Max-5 heap, and BST. Then you will repeat the runs many times and take the average for each operation and for each data structure.

You are also required to submit a written report discussing the following (please submit your report in PDF format):

1- Organization of experimental profiling.
2- Input data generated using a random number generator.
3- CPU time recording in C++.
4- Data recording and analysis.
5- Performance comparison, observations, and summary. (Experimentally determine the complexity of functions for each hash table and compare them to their theoretical complexities. If they are different, you need to explain why.)
6- Conclusions.

Array size: 10,000,000 (for heaps only)
m = 1,000,000

Steps:
1- Build (**using the bottom-up approach for heaps**) Min-5 heap, Max-5 heap, and BST by inserting the same set of random numbers with size m (the range should be from 1 to 5m). Using CPU time, record the build time for each heap and BST.
2- Delete the min value 0.001m times from the heaps and BST, and then record the CPU time for deleteMin for each heap and for BST. Then delete the max value 0.001m times from the heaps and BST and record the CPU time for deleteMax for each heap and BST.
3- Repeat steps 1 and 2 five times using different seeds and take the average.
4- Repeat steps 1-3 with different random numbers of size 2m, 3m, 4m, and 5m.

Random Number Generator:

```
#include <stdlib.h>
#include <time.h>

int num;
 /*initialize random seed*/
 srand(time(NULL));
 /*generate a random number between 1 and 10 inclusively*/
num = rand() % 10 + 1;
```

CPU Timing:

```
 #include<time.h>
clock_t t;
/*start the clock*/
t = clock();
 /*insert code to be timed here*/
t = clock() – t;
```

Expected Results:

File for testing the correctness of your BST, Min-5 Heap, and Max-5 Heap:
Data.txt: 21 5 4 10 22 2 32 44 10 30

Note: For option 1, you should give the level order for BST, Min-5 Heap, and Max-5 Heap exactly like you did for your Lab 5 and your Lab 8.

```
-----------------------------------------------------------
Please choose one of the following commands:
1- Test BST/Heaps
2- Performance Comparison
3- Exit
 >1

BST:
21 5 22 4 10 32 2 10 30 44

Min-5 Heap:
2
5 4 10 22 21
32 44 10 30

Max-5 Heap:
44
32 4 10 22 2
21 5 10 30
```

Performance (BST):

	1,000,000	2,000,000	3,000,000	4,000,000	5,000,000
Build(ms)					
deleteMin(ms)					
deleteMax(ms)					

Performance (Min-5 Heap):

	1,000,000	2,000,000	3,000,000	4,000,000	5,000,000
Build(ms)					
deleteMin(ms)					
deleteMax(ms)					

Performance (Max-5 Heap):

	1,000,000	2,000,000	3,000,000	4,000,000	5,000,000
Build(ms)					
deleteMin(ms)					
deleteMax(ms)					

Submission:

Follow the conventions below to facilitate grading:

Source Code

Place all your source files (*.cpp, *.hpp) and input files in a single folder with no subfolders.

- Name your folder using the convention Lastname_LabX (e.g., Smith_Lab11).
- Include a functioning Makefile inside the folder. (The makefile should also include the clean command.)
- **Verify that your code runs on the lab Linux machines before submission.**

Compressed File

- Compress using .zip, .rar, or .tar.gz.
- Name your file using the convention Lastname_LabX (e.g., Smith_Lab11.zip).

Email

- Use the following subject for your email: Lastname_LabX (e.g., Smith_Lab11).
- Send your code to l290w868@ku.edu if you are in one of Lei's sections or to dhwanipandya1401@ku.edu if you are in one of Dhwani's sections.
- Anytime you have a question about the lab, put the word question in the subject of the email.