Maximum possible marks: 100 Points

Due date:
11:59pm, Sunday 11/03/2019 for Monday lab.
11:59pm, Tuesday 11/05/2019 for Wednesday labs.
11:59pm, Thursday 11/07/2019 for Friday lab.

Purpose:
The purpose of this lab is to implement a min-leftist heap in C++.

General Requirements:
For this assignment, you will work on a pointer-based implementation of min-leftist heap. You are to read in the numbers from a data file of integers and insert each number into a min-leftist heap. Min-leftist heaps should not allow duplicate numbers.

For this lab, you should build the heap using the samples which are in the data.txt. After that, your program should have a simple menu like this:

------------------------------------------------------------
Please choose one of the following commands:
1- Insert
2- Merge
3- Deletemin
4- Findmin
5- Preorder
6- Inorder
7- Postorder
8- Levelorder
9- Exit

Min-leftist heap:
The min-leftist heap methods should be implemented as follows:
- Buildheap() - should build the min-leftist heap using the insert function.
- Insert(x) - should insert x into the min-leftist heap using the merge function. If the insertion causes the subtrees to be swapped, show that swapping has occurred as shown in the expected output illustration.
- Deletemin() - should delete the minimum value from the min-leftist heap and merge the remaining two sub heaps.
- Findmin() - should print the minimum value from the min-leftist heap.
- Merge(H1,H2) – this function should merge the two trees H1 and H2, where H1 and H2 are two min leftist heaps. As an example to be shown explicitly by your program to illustrate the working of this function, you are required to execute this function on two trees called H1 and H2, where H1, the first parameter, is the tree created by reading the data.txt file, and H2, the second parameter, is a tree created by taking 3 user inputs from
the command line, i.e., 88, 105, and 222. The new tree formed after merging is H3. Note that the task mentioned is only an example that is to be shown for the purpose of illustrating the proper working of this function through the menu. In general, this function should be called every time it is presented with two min leftist heaps and not only when it is called from the 2nd option of the menu.

- Preorder() should print the preorder traversal of the min-leftist heap.
- Inorder() — should print the inorder traversal of the min-leftist heap.
- Postorder() — should print the postorder traversal of the min-leftist heap.
- Levelorder() — should print the level order traversal of the min-leftist heap.

**Data file:**

data.txt: 21 15 24 38 25 5 55 87 71 14

We will insert these values, in the given order, into the min-leftist heap. The numbers in the boxes in the following diagrams indicate the rank. (Please refer to your lecture notes to obtain the formula to calculate the rank).

Insert 21

![Fig. 1](image1)

Insert 15

![Fig. 2](image2)

Insert 24

![Fig. 3](image3)
Insert 38

Fig. 4

Insert 25

Fig. 5

Swap the left and right subtrees of the root node since the rank of the right subtree is greater than the rank of the left subtree. After swapping, the tree should look like Fig. 6.

Fig. 6
Insert 5

After rest of the insertions, finally your tree should look like Fig.8

Expected output for the min-leftist heap:

Please choose one of the following commands:
1- Insert
2- Merge
3- Deletemin
4- Findmin
5- Preorder
6- Inorder
7- Postorder
8- Levelorder
9- Exit
>Enter your choice:
>2
>Output: Enter the three elements for the tree H2.
> 88
>105
>222
>Output: Trees H1 and H2 have been merged to form new tree H3. The new level order traversal for tree H3 is: 5 14 15 55 88 24 21 87 71 105 222 38 25.

------------------------------------------------------------
Please choose one of the following commands:
   1-  Insert
   2-  Merge
   3-  Deletemin
   4-  Findmin
   5-  Preorder
   6-  Inorder
   7-  Postorder
   8-  Levelorder
   9-  Exit

>Enter your choice:
>3
>Output: The minimum element has been deleted, and the tree has been rearranged. The new level order traversal is: 14 55 15 87 71 24 21 38 25 88 105 222.

------------------------------------------------------------
Please choose one of the following commands:
   1-  Insert
   2-  Merge
   3-  Deletemin
   4-  Findmin
   5-  Preorder
   6-  Inorder
   7-  Postorder
   8-  Levelorder
   9-  Exit

>Enter your choice:
>4
>Output: The minimum element is 14.
Please choose one of the following commands:

1) Insert  
2) Merge  
3) Deletemin  
4) Findmin  
5) Preorder  
6) Inorder  
7) Postorder  
8) Levelorder  
9) Exit  

> Enter your choice:  
> 1  
> Output: Enter the value you want to insert  
> 47  
> Output: 47 has been successfully inserted into the tree.  
> Subtrees have been swapped.  
> The new level order traversal is:  
> 14 15 55 24 21 87 71 38 25 88 47 105 222.

For menu options 5, 6, 7, 8 and 9 follow the same output structure as shown in Lab 07.

Questions:
Please answer the following question in not more than 5 lines each and submit it with your implemented code in PDF format.

1. Write the advantages and disadvantages of leftist heaps.
2. Explain the advantages and disadvantages of implementing a leftist heap using a pointer-based implementation instead of an array-based implementation.

Submission:
Follow the conventions below to facilitate grading:

Report
- Please include your answer (answer.pdf) in your folder before compressing it.

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_Lab08 (e.g., Smith_Lab08).
- 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_Lab08 (e.g., Smith_Lab08.zip).

Email
• Use the following subject for your email: Lastname_Lab08 (e.g., Smith_Lab08).
• Send your code to chiranjeevi.pippalla@ku.edu if you are in one of Chiru’s sections or to anubhav@ku.edu if you are in one of Anubhav’s sections.
• Anytime you have a question about the lab, put the word question in the subject of the email.