

**EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps**  
Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

**100 Points**

**Due date:**

11:59pm, Monday, 04/20/2020 for Tuesday labs

11:59pm, Wednesday, 04/22/2020 for Thursday labs

**Purpose:**

The purpose of this lab is implement and compare the performance of two priority queues: a min leftist heap and a min skew heap.

**General Requirements:**

In this lab, you are required to build two priority queue heaps - min leftist heap and min skew heap. Also, you will implement experimental profiling to compare the performance of the two heaps based on use of pointer-based implementations. The performance is to be compared for the build and delete min operations of each priority queue. The experiment is repeated with five different input data sizes. Each time, the CPU times for these two operations are recorded and compared. Below are the requirements and specifications. You should also plot the performance of the build function. In particular, **your plot should be of CPU time versus n, where n is the input size.**

**Data Specifications:**

**Input data:** There will be one input file (for example: data.txt) to test the heap. In addition, you are required to generate input data using a random number generator. This will give different sets of inputs generated with the below specifications:

**Random number generator:** Sample code is provided below.

```
.....  
  
#include <stdlib.h> // srand, rand  
....  
int random_number;  
  
srand (time(NULL)); // Initialize random seed: This means every time you run the program it will  
                     change the numbers generated.  
  
random_number = rand() % 10 + 1; // Generate random number between 1 and 10.  
  
// Execute the required functionality.  
  
.....
```

**Number of integers to input for the five data sets each with different seed:** There will be five different sets of inputs with sizes 100,000, 200,000, 300,000 400,000, and 500,000 simultaneously.

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

**Random number range:** Generate a sequence of random integers in the range 1 to 5,000,000 for the given input sizes.

**Sample input data file to test the data (i.e., data.txt):**

21 15 24 38 25

### Min Leftist Heap:

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

Insert 21

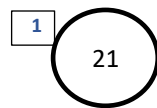


Fig. 1

Insert 15

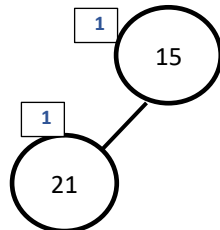


Fig. 2

Insert 24

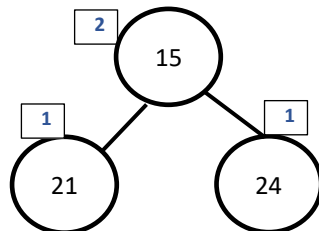


Fig. 3

Insert 38

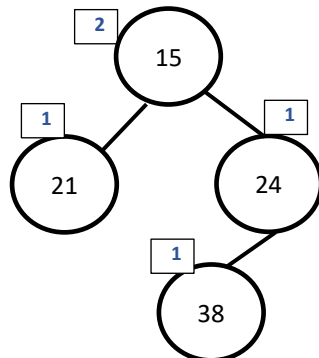


Fig. 4

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Insert 25

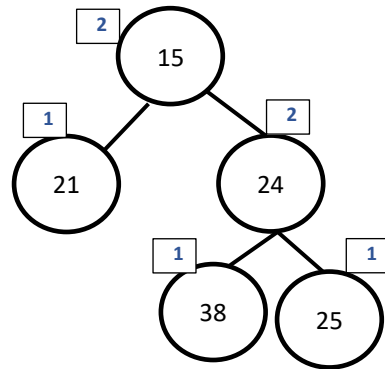


Fig. 5

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.

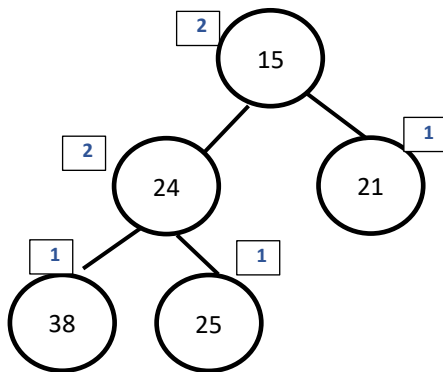


Fig. 6

**Delete min - Min leftist heap:** This will remove 15 which is minimum element in the heap.

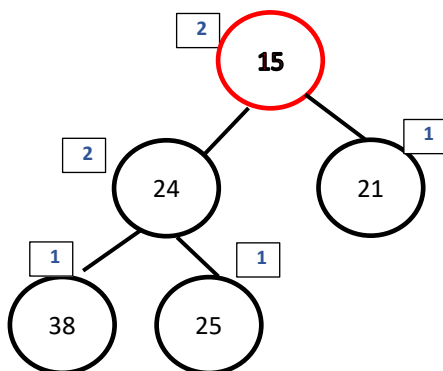


Fig. 7

This will partition the heap into two heaps as shown below.

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula



Fig. 8

Now these two heaps need to be merged. And then the left and right subtrees need to be swapped based on the rank.

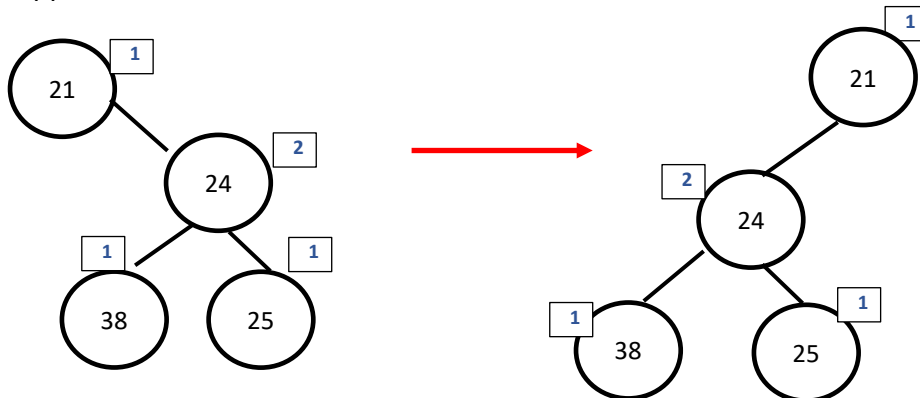


Fig. 9

**Min Skew Heap:** We will insert these values, in the given order, into the min skew heap.

Insert 21

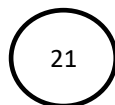


Fig. 10

Insert 15

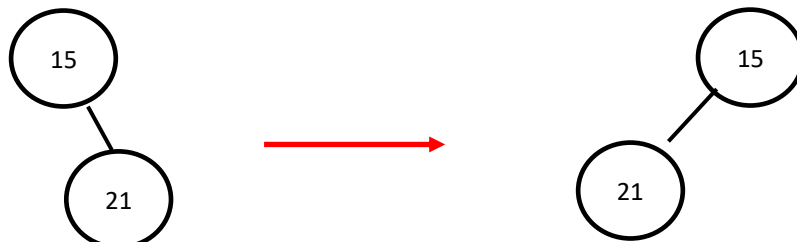


Fig. 11

While inserting 24, swap the left and right subtrees of the root node.

**EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps**  
Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Insert 24

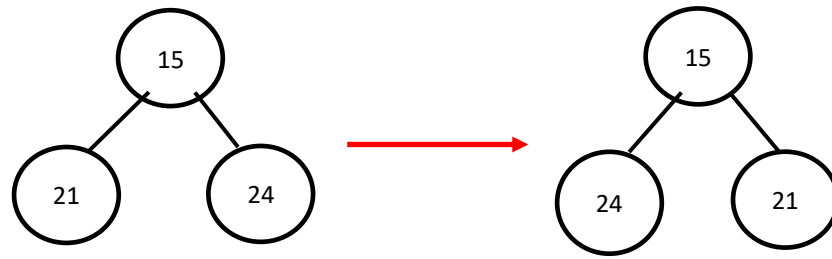


Fig. 12

While inserting 38, swap the left and right subtrees at each level until the root node is reached.

Insert 38

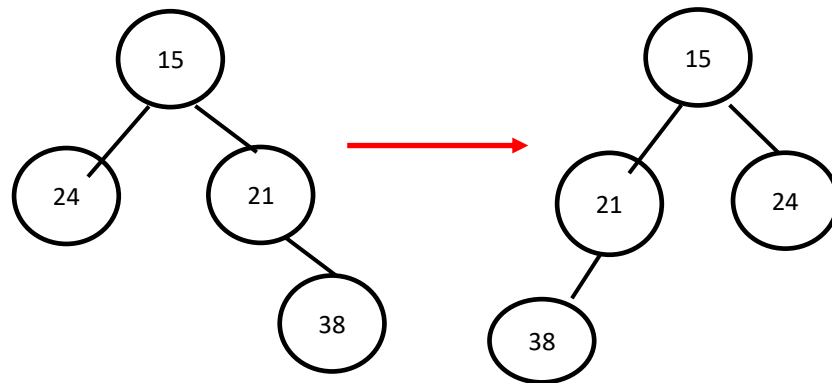


Fig. 13

While inserting 25, swap the left and right subtrees at each level until the root node is reached.

Insert 25

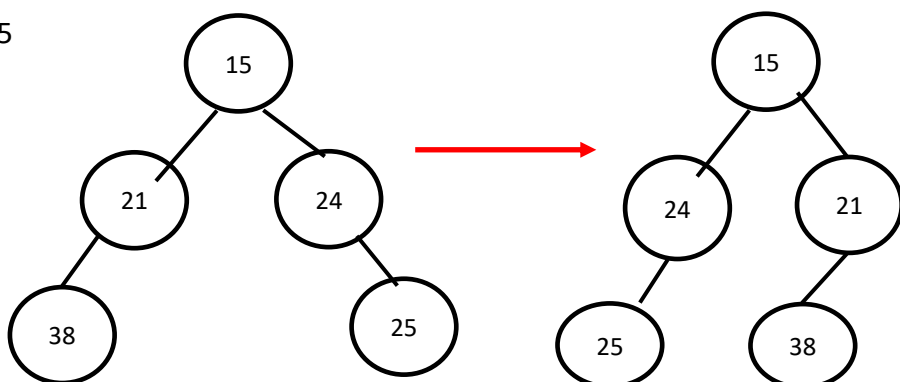


Fig. 14

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

**Delete min - Min skew heap:** This will remove 15 which is the minimum element in the heap.

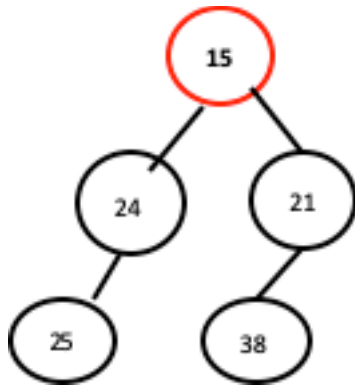


Fig. 15

This will decompose the heap into two heaps as shown below.



Fig. 16

Now these two heaps will be merged, and then the left and right subtrees need to be swapped as shown below.

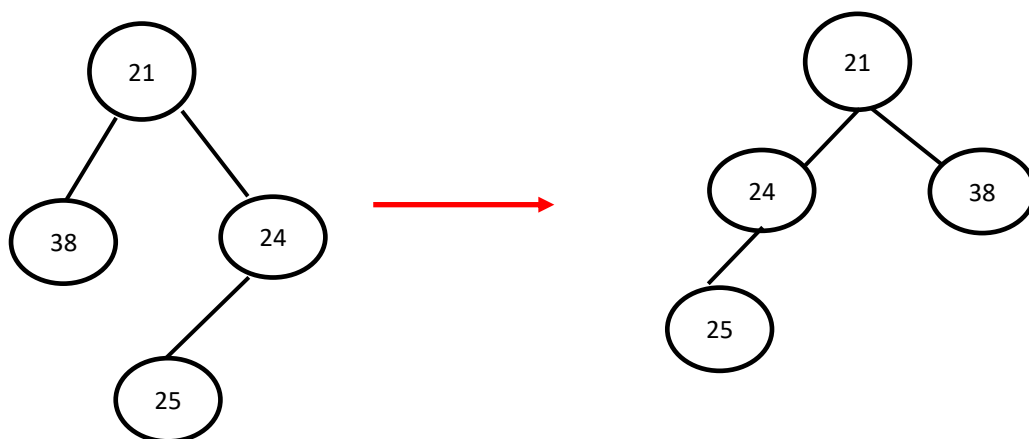


Fig. 17

**Execution Procedure:** As you execute the code, the program should show the below menu options. Provide the input file name (i.e., data.txt) while executing the code. Once the execution has started, the program should build the two heaps (i.e., the min leftist heap and min skew heap) and then show the menu as indicated.

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

-----Experimental profiling (Min leftist and Min Skew heaps)-----

1. Test build - min leftist heap
2. Test delete min - min leftist heap
3. Test build - min skew heap
4. Test delete min - min skew heap
5. Test performance of heaps
6. Exit

**Test build - min leftist heap():** This should show the result of the min leftist heap built from the input file (i.e., data.txt) which has been provided. The level order of the min leftist heap should be printed as output.

**Test delete min - min leftist heap():** This should delete the minimum element in the min leftist heap built from the input file (i.e., data.txt) which has been provided. That is the highest priority element (root) should be deleted, and after deletion, the level order of the min leftist heap should be printed as the result.

**Test build - min skew heap():** This should show the result of the min skew heap built from the input file (i.e., data.txt) which has been provided. The level order of the min skew heap should be printed as output.

**Test delete min - min skew heap():** This should delete the minimum element in the min skew heap built from input file. That is the highest priority element (root) should be deleted, and after deletion, the level order of the skew heap should be printed as the result.

**Test performance of heaps(): Note: This option does not use the input file (i.e., data file).** Once this option is selected, the random data should be generated. Then the performance of the heaps for the five different input sizes will be tested. Each input should have a different seed. The procedure outlined below should be followed:

1. Initialize the seed and generate the random data for the first input size (100,000) in the given data range above.
2. Start the CPU timer.
3. Build the min leftist heap.
4. Stop the CPU timer. Calculate and record the time (in seconds) required to build the min leftist heap.
5. Start the CPU timer.
6. Delete the minimum element in the min leftist heap.
7. Stop the CPU timer. Calculate and record the time (in seconds) required to delete the minimum element in the min leftist heap.
8. Start the CPU timer.
9. Build the min skew heap.
10. Stop the CPU timer. Calculate and record the time (in seconds) to build the min skew heap.
11. Start the CPU timer.
12. Delete the minimum element in the min skew heap.

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

13. Stop the CPU timer and calculate and record the time (in seconds) required to remove the minimum element from the min skew heap.

Repeat the above procedure for all the other input sizes given above (i.e., 200,000, 300,000, 400,000, and 500,000). The expected results are shown below.

### Expected Results:

Using sample input file data.txt: 21 15 24 38 25

The following outputs are just for illustrative purposes only. Sample output is given below and as shown in the expected output. The timing should only be recording while testing the performance of the heaps (option 5 in the menu). As the execution starts, the program should build the min leftist heap and the min skew heap from the input file given (i.e., not from the random numbers). The system should display the menu below.

-----Experimental profiling (Min leftist and Min Skew heaps)-----

1. Test build - min leftist heap
2. Test delete min - min leftist heap
3. Test build - min skew heap
4. Test delete min - min skew heap
5. Test performance of heaps
6. Exit

Please choose an option:

1

Level order of min leftist heap:

15 24 21 38 25

-----Experimental profiling (Min leftist and Min Skew heaps)-----

1. Test build - min leftist heap
2. Test delete min - min leftist heap
3. Test build - min skew heap
4. Test delete min - min skew heap
5. Test performance of heaps
6. Exit

Please choose an option:

3

Level order of min skew heap:

15 24 21 25 38



## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

-----Experimental profiling (Min leftist and Min Skew heaps)-----

1. Test build - min leftist heap
2. Test delete min - min leftist heap
3. Test build - min skew heap
4. Test delete min - min skew heap
5. Test performance of heaps
6. Exit

Please choose an option:

2

Minimum element deleted

Level order of min leftist heap:

21 24 38 25

-----Experimental profiling (Min leftist and Min Skew heaps)-----

1. Test build - min leftist heap
2. Test delete min - min leftist heap
3. Test build - min skew heap
4. Test delete min - min skew heap
5. Test performance of heaps
6. Exit

Please choose an option:

4

Minimum element deleted

Level order of min skew heap:

21 24 38 25

-----Experimental profiling (Min leftist and Min Skew heaps)-----

1. Test build - min leftist heap
2. Test delete min - min leftist heap
3. Test build - min skew heap
4. Test delete min - min skew heap
5. Test performance of heaps
6. Exit

Please choose an option:

5

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

### Performance (Min Leftist Heap)

Input size	100,000	200,000	300,000	400,000	500,000
Build (s)	0.012568	0.0214294	0.0340122	0.0477508	0.0624752
Delete min (s)	0.012568	0.0214294	0.0340122	0.0477508	0.0624752

### Performance (Min Skew Heap)

Input size	100,000	200,000	300,000	400,000	500,000
Build (s)	0.012568	0.0214294	0.0340122	0.0477508	0.0624752
Delete min (s)	0.012568	0.0214294	0.0340122	0.0477508	0.0624752

-----Experimental profiling (Min leftist and Min Skew heaps)-----

1. Test build - min leftist heap
2. Test delete min - min leftist heap
3. Test build - min skew heap
4. Test delete min - min skew heap
5. Test performance of heaps
6. Exit

Please choose an option:

6

Bye Bye!!

### Report:

Along with the lab submission, you should submit the plot that is created based on the build time for each input size. You should submit two plots for each heap - one for min leftist heap and one for min skew heap. These plots should be based on your results from the lab experiments.

### Grading rubric:

- Full grade: The program should execute without any issues with all the options executed and with no memory leaks.
- Points will be taken off for execution errors, such as memory leaks, segmentation/program abort issues and missing handling of invalid cases.

## EECS 560 Lab 9 – Experimental Profiling on Leftist and Skew Heaps

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

- Programs that are compiled but do not execute will earn in the range of 0 to 50% of the possible points. Your grade will be determined based on the program design and the options implemented in the code.

### Submission instructions:

- All files, i.e., the source files and Makefile, should be zipped in a folder.
- Include a ReadMe.txt if your code requires any special instructions to run.
- The naming convention of the folder should be LastName\_Lab9.zip (or .tar or .rar or .gz).
- Email it to your respective lab instructor: [chiranjeevi.pippalla@ku.edu](mailto:chiranjeevi.pippalla@ku.edu) (Chiru) or [prashanthi.mallojula@ku.edu](mailto:prashanthi.mallojula@ku.edu) (Prashanthi) with subject line EECS 560 Lab9.
- Your program should compile and run on the **Linux machines** in **Eaton 1005D** using **g++**.