EECS 560 Lab 6 – Implementation of Min-max Heap

Prof.: Dr.Shontz, GTAs: Sirisha Thippabhotla, Prashanthi Mallojula

Maximum possible marks: 100 Points

Due date:
03/29/2021, 08:00 am - Monday 10:00 am Lab
03/29/2021, 12:00 pm - Monday 02:00 pm Lab
03/31/2021, 09:00 am - Wednesday 11:00 am Lab
03/31/2021, 12:00 pm - Wednesday 02:00 pm Lab

Purpose:
For this lab, you will implement a min-max heap in C++.

General Requirements:
In this assignment, **you will develop an array-based implementation** of a min-max heap. **The initial build of the min-max heap should use the top-down approach**. Duplicates are allowed to be inserted. Also, each time you insert/remove an element, you should first seek to maintain a complete tree (structural property of the heap) and then reorder the nodes to satisfy the min-max heap-ordered tree property.

In the min-max heap:

The root of T will be on a minimum level, and the next level will be a maximum level. Minimum and maximum levels will alternate until all the records in the data.txt file are inserted into the heap.

Here is where the max and min nodes are located:
min nodes: even levels (0, 2, 4…)
max nodes: odd levels (1, 3, 5…).

The min-max heap methods should be implemented as follows:

- BuildHeap() - Should build the min-max heap using top-down approach.
- Insert(x) – Should insert x into the min-max heap. This means, you should add a new element into the heap. After the insertion of new element, the heap should still satisfy both the structural property (i.e., a complete tree) and the min-max heap-ordered property.
- Delete() – Should delete the root element from the min-max heap. After deleting the element, the heap should still satisfy both the structural property (i.e., a complete tree) and the min-max heap-ordered property.
- MinLevelElement() – Should print out the elements of the min-max heap at even levels in level order.
- MaxLevelElements() - Should print out the elements of the min-max heap at odd levels in level order.
- Exit – should exit from the program.

EECS 560 Lab 6 – Implementation of Min-max Heap

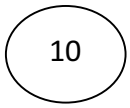
Prof.: Dr.Shontz, GTAs: Sirisha Thippabhotla, Prashanthi Mallojula

The file you need to read in the input will be 'data.txt'. You are allowed to hard code the file name in the program.

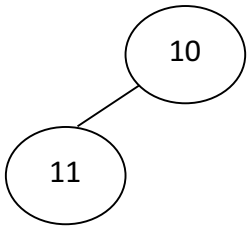
data.txt:
10, 11, 5, 13, 19, 22, 9, 8, 25, 7, 2

Let's look at how the min-max heap works diagrammatically. Here, the nodes are inserted into the heap in a level-order fashion, to maintain a complete tree.

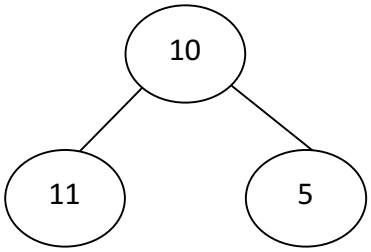
Step1: Insert 10



Step2: Insert 11

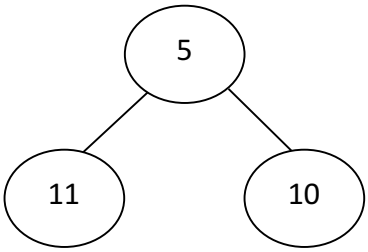


Step3: Insert 5

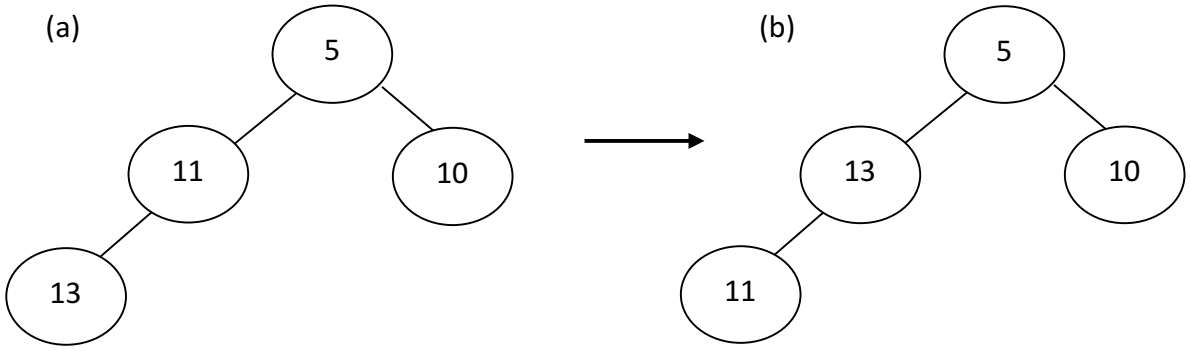


In Step 3, the min-max heap property is violated. 10 is not smaller than 5, so we swap 5 with 10.
Note: The element at the root will always be the smallest element.

Step 4: Satisfying the min-max heap property

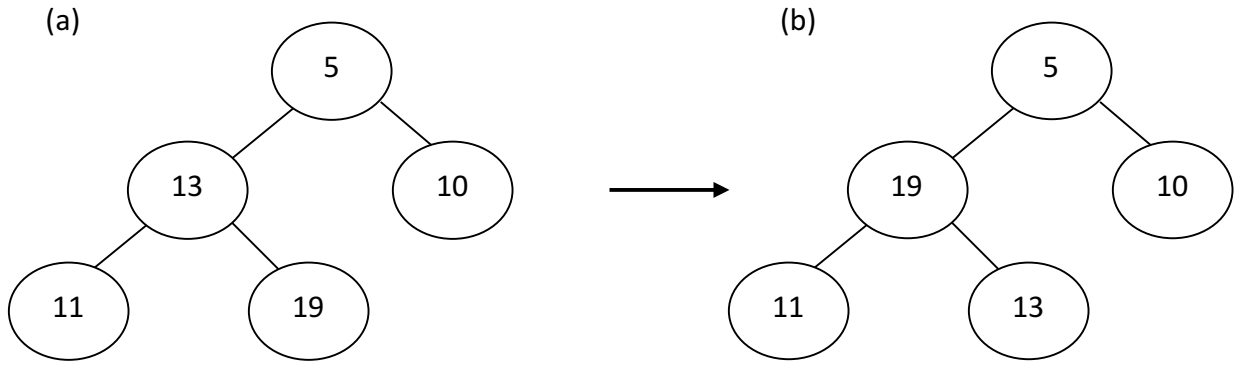


Step 5: Insert 13



When 13 is inserted as the left child of 11 (fig 5a), 13 will be greater than 11. So, we swap the positions of 13 and 11 (fig 5b).

Step 6: Insert 19

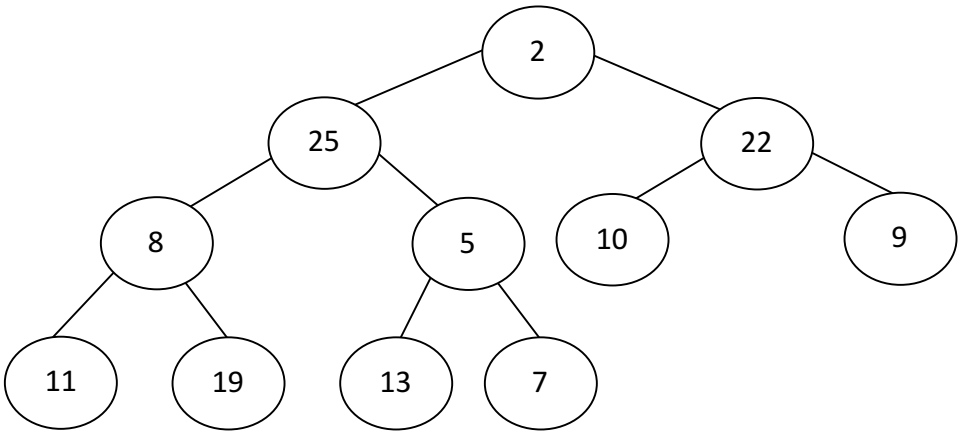


When 19 is inserted as the right child of 13, 19 is greater than 13 (fig 6a). So, 19 and 13 will be swapped. The diagram to the right shows the heap after inserting 19 and swapping it with 13 (fig 6b).

Final Step: The below diagram is the final min-max heap representation after inserting all the elements and heapifying until the min-max heap-ordered property is satisfied.

EECS 560 Lab 6 – Implementation of Min-max Heap

Prof.: Dr.Shontz, GTAs: Sirisha Thippabhotla, Prashanthi Mallojula



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

Please choose one of the following commands:

1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>1
>Output:
2,
25, 22,
8, 5, 10, 9,
11, 19, 13, 7

Please choose one of the following commands:
1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:

>2
>Enter element to be inserted: 23
>Output: 23 has been inserted successfully.

Please choose one of the following commands:
1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>5
>Output:
25, 23,
11, 19, 13, 7, 22

Please choose one of the following commands:
1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>3
>Output: 2 has been deleted successfully.

Please choose one of the following commands:
1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>4
>Output:

EECS 560 Lab 6 – Implementation of Min-max Heap

Prof.: Dr.Shontz, GTAs: Sirisha Thippabhotla, Prashanthi Mallojula

5,
8, 7, 10, 9

Please choose one of the following commands:
1- BuildHeap
2- Insert
3- Delete
4- MinLevelElements
5- MaxLevelElements
6- Exit

>Enter your choice:
>6
>Output: Bye Bye!

Grading rubric:

- Full grade: The program should execute without any issues with all the options executed and with no memory leaks.
- 40% grade deduction: The STL (Standard Template Library) was used. **(Please refer to the lab policy on not using the STL.)**
- Points will be taken off for execution errors, such as memory leaks, segmentation/program abort issues and missing handling of invalid cases.
- 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_Lab6.zip (or .tar or .rar or .gz).
- Submit your lab via Blackboard through the submission box created for Lab6.
- Your program should compile and run **using ssh on the EECS cycle servers.** (We will be testing it solely on these servers.)