EECS 560 Lab 8 – Implementation of Maxmin Heap

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

Maximum possible marks: 100 Points

Due date:
11:59pm, Monday 04/13/2020 for Tuesday lab.
11:59pm, Wednesday 04/15/2020 for Thursday lab.

Purpose:
For this lab, you will implement a Maxmin heap in C++.

General Requirements:
In this assignment, **you will develop an array-based implementation** of a Maxmin heap. **The initial build of the** Maxmin **heap should use the top-down approach**. Duplicates are allowed to be inserted. Also, each time you insert/remove an element, you should organize the heap following the Maxmin heap property.

In the Maxmin heap:

The root of T will be on a maximum level, and the next level will be a minimum level. Maximum and minimum 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:
max nodes: even levels (0, 2, 4…)
min nodes: odd levels (1, 3, 5…).

The file you will read the items from will be data.txt. The input file consists of the records. Each record consists of two fields – the Google play store app name and the number of downloads in the 1000s. The number of downloads can be huge in number, for example 60K. You need to consider the integer before K. For example: fruitninja, 60K. Your data.txt file should consider only fruitninja and 60 while building the heap. Each record in the input file will have the following format: game name (type: string), number of downloads (type: integer).

The Maxmin heap methods should be implemented as follows:

- AddGame - should insert a new game into the Maxmin heap. This means you should add a new game record into the heap. After the insertion of new record, the heap should still satisfy the Maxmin heap property.
- DeleteMaxDownloadedGame - should delete the game record with the maximum number of downloads from the Maxmin heap. In our case, it will be the root node. After deleting the record, the heap should still satisfy the Maxmin heap property. Also note that if there is a duplicate record that also has the same number of maximum downloads, you should only delete the record in the root node.
- DeleteGame - should delete the game record with the particular number of downloads from the Maxmin heap. The user should see a prompt asking for the game name and

the number of downloads. When the inputs are given, that particular node has to be deleted from the heap. After deleting the record, the heap should still satisfy the Maxmin heap property. Also note that if there are two games with an equal number of downloads, then using the name of the game as an additional feature, find out the game the user wants to delete and then delete the game from the heap. Even if the name and number of downloads match, then the first occurrence of the record has to be deleted.

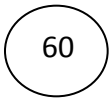
- PrintGamesAtMinimumLevels - should print the names of games at minimum levels in the Maxmin heap, i.e., games at odd levels in the Maxmin heap in level order.
- PrintGamesAtMaximumLevels - should print out the names of games at maximum levels in the Maxmin heap, i.e., games at even levels in the Maxmin heap in level order.
- TotalMinimumDownloadedGames – should print the sum of all the minimum downloaded games in the Maxmin heap.
- TotalMaximumDownloadedGames – should print the sum of all the maximum downloaded games in the Maxmin heap.
- Exit – should exit from the program.

The data.txt file will look like below.

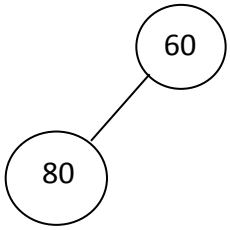
fruitninja,60
subwaysurfers,80
roadrash,50
pubg,20
ninjajump,70
vector,80
sudoku,20
templerun,90

Example: 60, 80, 50, 20, 70, 80, 20, 90
Let's look at how the Maxmin Heap works diagrammatically.

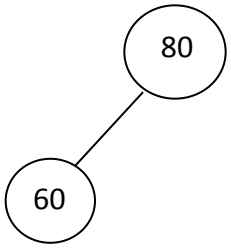
Step1: Insert 60



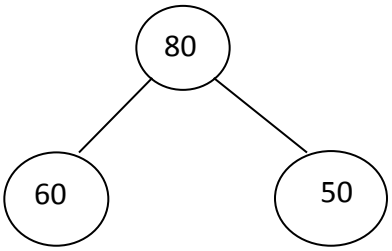
Step2: Insert 80



Here the Maxmin heap property is violated. So swap 60 with 80.

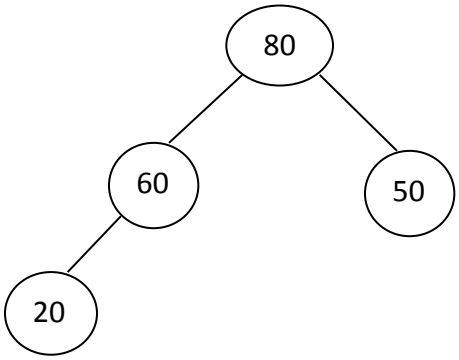


Step3: Insert 50



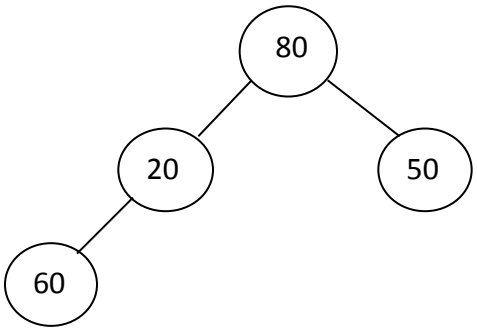
Note: The element at the root will always be the largest of the elements.

Step 4: Insert 20

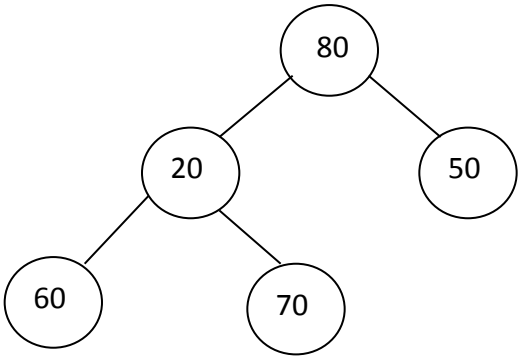


Here the maxmin property is violated. So swap 60 and 20.

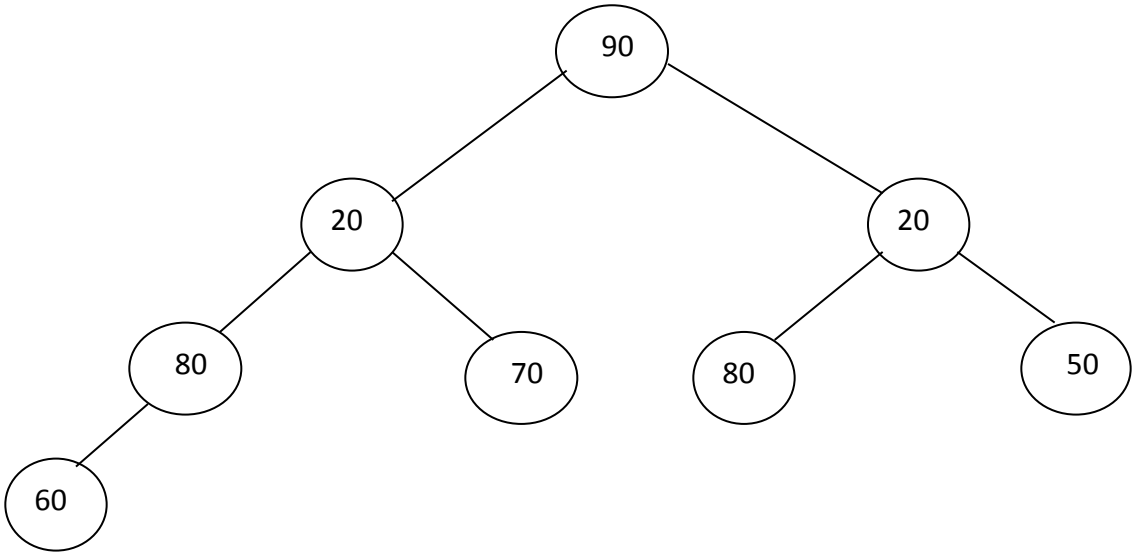
Note: When a new element is added. It has to be first compared with its parent and check if it is satisfying the local maxmin property.



Step 5: Insert 70



Final Step: The below diagram will be the final Maxmin Heap representation after inserting 80, 20, and 90 from the above example are inserted following the Maxmin property.



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:

EECS 560 Lab 8 – Implementation of Maxmin Heap

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

---

Please choose one of the following commands:

1- AddGame
2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>1
>Enter the name of the game you want to insert into the play store:
>candycrush
>Enter the number of downloads for the game in thousands:
>120
> Output: A new game was successfully added to the playstore.

Please choose one of the following commands:

1- AddGame
2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMaximumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>1
>Enter the name of the game you want to insert into the play store:
>angrybirds
>Enter the number of downloads for the game in thousands:
>10
>Output: A new game was successfully added to the playstore.

Please choose one of the following commands:

1- AddGame
2- DeleteMaxDownloadedGame

3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>2
>Output: candycrush game with 120K downloads has been deleted successfully.

Please choose one of the following commands:
1- AddGame
2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>4
> Output: angrybirds, pubg, fruitninja, sudoku

Please choose one of the following commands:
1- AddGame
2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>5
> Output: templerun, vector, ninjajump, subwaysurfers, roadrash

---
Please choose one of the following commands:
1- AddGame

2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>6
> Output: 40

Please choose one of the following commands:
1- AddGame
2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>7
> Output: 90

Please choose one of the following commands:
1- AddGame
2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- TotalMinimumDownloadedGames
7- TotalMaximumDownloadedGames
8- Exit

>Enter your choice:
>3
>Enter the name of the game you want to delete:
>fruitninja
>Enter the number of downloads of the game you want to delete:
>60

EECS 560 Lab 8 – Implementation of Maxmin Heap

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

>Output: fruitninja game with 60K downloads has been successfully deleted.

Please choose one of the following commands:

1- AddGame
2- DeleteMaxDownloadedGame
3- DeleteGame
4- PrintGamesAtMinimumLevels
5- PrintGamesAtMaximumLevels
6- MinLevelGames
7- MaxLevelGames
8- Exit

>Enter your choice:
>8
> Output: Bye!!!

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.
- 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_Lab8.zip (or .tar or .rar or .gz).
- Email it to your respective lab instructor: chiranjeevi.pippalla@ku.edu (Chiru) or prashanthi.mallojula@ku.edu (Prashanthi) with subject line EECS 560 Lab8.
- Your program should compile and run on the **Linux machines** in **Eaton 1005D using g++.**