# California State University, Fresno

# DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | **Algorithms & Data Structures** | | | Semester: | **Fall 2020** |
|  | | | | | |
| Points |  | Document author: | **Yet Chun Fong** | | |
|  | Author’s email: | **Fongyc7177 @mail.fresnostate.edu** | | |
| Laboratory number: | **4** | | |
|  | | | | | |

**1. Statement of Objectives**

The main objective of this lab session is to gain experience in the functions of both binary heaps and priority queues. Binary heap is much preferred for implementation of priority queue because heaps provide a much better performance compared to arrays to linked lists. Binary heaps only take O(log N) time for insertion. Furthermore, multiple functions have been implemented for priority queue to insert, delete, and modify nodes of priority queue without opposing the heap properties. Not only that, another task for this lab session is to sort a list of numbers in ascending/descending order using Max-Heap/Min-Heap.

In this lab session, the differences of Min-Heap and Max-Heap is covered. Min-Heap has the smallest value on top of the tree, also known as root. On the other hand, Max-Heap has the largest value as the root node. In addition, the parent node is always larger than their children in Max-Heap. For Min-Heap, the parent node is always smaller than their child node. Both Min-Heap and Max-Heap are complete binary tree which all leaves are on the same level, and nodes are inserted from left to right and delete from right to left. In order to, find the index of parent of a node, (i-1)/2 can be used. On the other hand, calculation of (2\*i)+1 and (2\*i)+2 can be used to find left child and right child of a node.

**2. Experimental Procedure**

First, a class is created to implement the Max-Heap and Min-Heap function. In the private section, a vector has been declared and basic functions to return the parent and children nodes are placed under private. Moving to public, the first function created is print function, to print all element in the vector. Then, Max Heapify function has been created to make the list of number in vector to be sorted with Max-Heap property and followed by a build Max Heap function. The next function is Ascending which is used to sort the vector in ascending order with Max-Heap properties. Then, insert function for Max-Heap is created which will push the new value into the vector and Max-Heapify function is called to make sure every node inserted into the binary heap tree will fulfill the heap properties. Other functions such as modifyMax(), extractMax(), and getMax() are being implemented under Max-Heap. extractMax() is used to delete the largest value, which is the root node for Max-Heap, and getMax() is just returning the largest value in the tree, the root node but not delete it.

Similar functions are being created for Min-Heap but in the opposite way because the smallest value has to be the root of the heap tree to comply with Min-Heap properties. Then, Descending function is being implemented under Min-Heap to sort the vector in descending order under Min-Heap properties. The extractMin() will delete the root node under Min-Heap properties and getMin() will be used for getting the value of root node; the smallest value but not deleting it.

In the main function, a vector named test is being created to be used for both Max and Min heaps and to be sorted in ascending and descending orders. Functions like insert, extract and print are being executed as well.

**3. Analysis**

Text

Description automatically generated

The order of results is based by the Sample Output word document. When user enter 0, the ascending order function (Max-Heap) will be run. If the user enters 1, the descending order (Min-Heap) function will be executed. For Max-Heap, the first function executed is insertion, when the value of 9 is being inserted into the vector, the ascending sorted list is 2,3,4,6,9. When the extractMax() function is being run, the value 9 is removed from the vector because it is the largest value in the Max-Heap tree, so the final sorted output will be 2,3,4,6. Similar to Min-Heap, when the insertion function is being executed, the descending sorted heap is 9,6,4,3,2. Then, when the extractMin() in Min-Heap is being run, the value 2 is removed from the vector because it is the smallest value in the Min-Heap tree, the final output is 9,6,4,3. Lastly, user can exit the program by typing 2 and the program will stop

**4. Encountered Problems**

There is a significant issue during this lab session. There is a ‘Debug Assertion Failed’ error message pop up after each run of program which is caused by the printAll() function. In addition, “Expression: vector subscript out of range” is also being shown in the message; sounds like the problem of stack overflow. Multiple changes have been made in attempt to resolve this issue. For instance, the error message will not pop up if reduce the size of vector in print function will not print the correct output. The final output of Max-Heap should be 2,3,4,6 but 2,3,4 will be the output if the size of n has been reduced by 1. No message of compilation error is being shown. Then, the second method has been used which changing the function of insert. There is a line of code which is n=n+1 in the insertion function for both Max-Heap and Min-Heap, and that is the main problem of this error because the size of vector has been modified, as the new value is being push into the vector. After removing that line of code in insertion function. The problem has been resolved, and the program is able to run correctly.

**5. Conclusions**

After this lab session, it is being proven that the binary heap is a much better implementation for priority queue compared to array or linked list because the properties of binary heap, either Max-Heap or Min-Heap are compatible with the properties and functions of priority queues, and provide a much better performance in terms of time complexities. In addition, Max-Heap can be used to sort a list of number in ascending order and Min-Heap can be used to sort a list of number in descending order by swapping the last node and root node, then reduce the size of heap, and heapify the tree. This process continue recursively until the size of heap size is being reduced to 1. It is also being proven that extract or return the largest/smallest value in Max-Heap/Min-Heap only take O(1) time because it is the root node, also known as the first element in the binary heap tree.

**6. References**

Coding Guidelines provided by Lab TA

Lecture slides (L09- Heap Sort)