

335 project 3

due date: July 14th, 2024

Priority Queue

1 Task Description:

This assignment is suppose to simulate customers arriving at a problems store(a problems store is a store that will solve random problems customers might have). When the customer arrives at the problems store his "problem" is a given a priority level(depending on how important the problem might be). Customers are then served by the store's clerk based on their priority level. Once the clerk is finished with a customer, he will record how long it took to solve that customers problem(service time), and will keep a record of all the customers that visited the store, with their respective service time. At the end of the day the clerk sorts the record of customers based on their service time.

2 To Do List:

1. Create a template class for a MaxHeap by modifying the code given in the notes for MinHeap. Property of a MaxHeap: a binary tree has the heap order property if every node is bigger than or equal to its two children.
2. Create a class(or struct) for Customer, with member variables string name_, int service_time_, and int priority_level_.
3. Implement a member function for Customer, SetPriorityLevel(), that uses the rand() function from `#include<cstdlib>`, to set priority_level_ to a random integer between 0 and 100. Make sure to call this function in your constructor.
4. Implement another member function, SetServiceTime(), that sets service_time_ to a random integer between 0 and 60. Do not call this function in your constructor.
5. Modify the Customer class you created so that it uses priority_level_ as the comparable metric, when inserted into the MaxHeap.
6. Modify the MaxHeap template class function DeleteMax() to return the Customer object being deleted.

7. In a main.cpp, run the simulation! create a bunch of random customers and insert them into a MaxHeap. (customers arriving at store)
8. (Still in main.cpp)Run a for loop to remove all the elements of the MaxHeap(the clerk servicing the customers), when a Customer is removed from the heap, the object being removed should update its service time(use member function from part 2 SetServiceTime()) and moved to vector<Customer> history(the record of customers kept by clerk)
9. (Still in main.cpp)Sort history based on service_time_ in ascending order.