Turnitin Originality Report

Processed on: 31-Dec-2023 22:40 +08
ID: 2265822146
Word Count: 881
Submitted: 2

Assignment 2 By Wan Nur Sofea Mohd Hasbullah

Similarity Index	Similarity by Source	
4%	Internet Sources:	2%
	Publications:	0%
	Student Papers:	4%

2% match (student papers from 13-Oct-2023)
Submitted to University of Lincoln on 2023-10-13

2% match (student papers from 02-Oct-2023)
Submitted to The NorthCap University, Gurugram on 2023-10-02

Department of Computer Science Faculty of Computing ASSIGNMENT 2 : APPLY DATA STRUCTURE OPERATIONS - LINKED LIST DATA STRUCTURE AND ALGORITHM SECJ 2013 - 02 CASE STUDY: RESTAURANT MANAGEMENT SYSTEM (SYSTEM TO ORDERS IN A RESTAURANT) GROUP: TUPPERWARE NO. NAME MATRIC NUMBER 1 WAN NUR SOFEA BINTI MOHD HASBULLAH A22EC0115 2 NADHRAH NURSABRINA BINTI ZULAINI A22EC0224 3 NUR ALEYSHA QURRATU'AINI BINTI MAT SALLEH A22EC0241 TABLE OF CONTENT 1.0 INTRODUCTION 3 1.1 Objective of The Project 3 1.2 Synopsis Project 3 2.0 SYSTEM DESIGN 4 2.1 Class Diagram 4 2.2 Pseudocode 5 2.3 Implementation of Linked List 7 2.4 Implementation of Sorting 8 2.4 Implementation of Delete 9 1.0 INTRODUCTION 1.1 Objective of The Project • To create a system where users can insert, delete and find the menu they want from the system. • Allow users to sort the list menu in ascending or descending order. • Allow users to display the list of the menu. 1.2 Synopsis Project The type of data structure used in the Restaurant Management System is a linked list. We added another two classes, which are class Node and class OrderList. The purpose of the class Node is to point to the content of the next node and the data in the linked list. The OrderList is a class that contains a head and a pointer to the first node in the list. Head is set to null since the list is empty initially. The system allows users to insert a new menu and delete the menu either at the beginning, middle or back of the list. Furthermore, users can also find a menu from the menu list by entering the name of the menu, category of the food, or food ID. Additionally, the system allows users to sort the menu in ascending or descending order. Lastly, users can ask the system to display all the menu items in the list. 2.0 SYSTEM DESIGN 2.1 Class Diagram 2.2 Pseudocode 2.3 Implementation of Linked List This assignment requires a few manipulations of the node using a linked list, such as inserting a new node and deleting a node at a time. In the insert new node, four manipulating operations have been implemented: insert a new node at the front(insertAtFront), at the end of the list(insertAtEnd) and at the middle of the list(insertAtMiddle). For the insertFront function, it will insert a new node at the very first node in the list. The new node will be the head while previous head will become the second node. Meanwhile, insertAtEnd function only add a new node at the very last node in the list. The funciton will not modify any node before as there is only an addition of node. Lasty, insertAtMiddle function is to add new node at any place in the list. This function asks user to input a food ID that exist in the list. When the system read the food ID, the function will add the new node after the node inserted by user. For the delete node, there are three operations which are delete at the front of the list(deleteFront), at the end of the list (deleteEnd) and delete at the middle of the list (deleteMiddle). The deleteFront function operates by deleting the first node in the list. The deleteEnd function is used to delete the last node in the list. Furthermore, the deleteMiddle function operates by deleting a node from the middle of the node list. These three operations will print out an error message if the list is empty. The node list will be updated after delete node is operated. There are three different methods to find a node: findNodeName, findNodeCategory, and findNodeFoodId. These methods are implemented in the OrderList class to search menus based on name, food category and food id. Through the linked list, these operations compare the search criteria with the menu attributes of each node until a match is found. 2.4 Implementation of Sorting This coding also implements a sorting operation to give users a better viewing experience. The selection concept is used again, with a time complexity of O(n2), to sort the menu list based on two users' input: food ID or price. Compared to the previous assignment, this assignment implements sorting in both ascending and descending. Giving users a variety of options to choose from. This sorting algorithm differs from the previous coding, taking the Orderlist variable as the parameter. Both functions, sortFoodID, and sortPriceID, will prompt the user to enter either 1 or 2. If the user input 1, the boolean parameter being passed on is true, while for input 2, the parameter is false. The ascending order will be operated on if true, meanwhile descending on false. The condition will first check whether the current node(the head) is null. If the head is empty, a minMaxNode is assigned to the current node, and a temporary node is given to the next node for the current. An inner looping is performed to find the maximum or minimum node in the list. Every time a new node has a smaller or bigger value, the minMaxNode will be updated. Swaps will occur after the maximum or minimum node is acquired until all nodes are in the correct position. Finally, the function will return the newly sorted menu list.