Turnitin Originality Report

Processed on: 20-Dec-2023 22:27 +08
ID: 2263146504
Word Count: 1229
Submitted: 1

Assignment 1 By safwan Azmi

Similarity Index	Similarity by Source	
3%	Internet Sources:	2%
	Publications:	0%
	Student Papers:	2%

2% match (Internet from 26-Oct-2023)
https://ifacet.iitk.ac.in/searching-and-sorting-algorithms/

1% match (student papers from 26-Nov-2023)
Submitted to Kaplan International Colleges on 2023-11-26

Department of Computer Science Faculty of Computing Assignment 1 (Sorting & Searching Process) [13] Courier Service System Programme Subject Code Subject Name Session-Sem : Bachelor of Computer Science (Data Engineering) : SECJ2013 : Data Structure & Algorithms : 2023/2024-1 Prepared by : 1) MUHAMMAD DANIEL HAKIM BIN SYAHRULNIZAM (A22EC0207) 2) MUHAMMAD NUR AZHAR BIN MOHD YAZID (A22EC0220) 3) MUHAMMAD SAFWAN BIN MOHD AZMI (A22EC0221) Section : 02 Group : TechTurtles Lecturer : Dr. Lizawati Md Yusuf 1. Objective The Courier Service System's main objective is to implement sorting and searching into the program. The courier service has tons of parcels in the inventory, so the system needs to have great efficiency in handling the business of parcels, mostly in terms of sorting and searching to ease the system user's view and understanding. The searching method to search a parcel needs to minimize errors and time taken as possible. Also, the time taken to sort the parcels by its delivery status, shipping option, tracking number needs to be as minimal as possible, so we need to think through the best sorting solution for the system to utilize. 2. Synopsis The Courier Service System is specially designed for the administrator. The system's objectives are mainly to sort parcels efficiently based on their delivery status, shipping option or tracking number by implementing bubble sort, merge sort and selection sort respectively. Moreover, it also helps the administrator to add a new parcel into the system by implementing binary search. Not to mention, the admin will be able to search the intended parcel based on the parcel's tracking number. 3. Design (class design presented in a class diagram and/or algorithm design illustrated in pseudocode or a flow chart) Class Diagram Pseudocode 1. Start 2. The system will display a welcome message. 3. The system will provide five options to the user: 3.1. If the user chooses case '1', the user will choose to add a new parcel to the system. 3.1.1. Function addNewParcel will be executed with parcel and count as arguments. 3.1.2. The user will be prompted to input the parcel's tracking number, address, sender's name, receiver's name, shipping option, and delivery status. 3.1.3. End if 3.2. If the user chooses case '2', the user wants to search for a particular parcel by using parcel tracking number as the input. 3.2.1. The user will be prompted to input the parcel's tracking number. 3.2.2. Function binarySearch will be executed with parcel, tracking number, and count as arguments. 3.2.3. The system will

search the parcel by making a comparison between input and the existing parcel's tracking number using binary search. 3.2.3.1. If a match is found, the system will display the parcel's details. 3.2.3.2. Else, the system prompts the user to try again. 3.2.3.3. End if 3.2.4. End if 3.3. If the user chooses case '3', the user will prompt the system to sort the parcels' delivery status (complete or incomplete). 3.3.1. The system will give the user two options: either to sort in ascending or descending order. 3.3.1.1. If the user enters '1', the delivery status of the parcel will be sorted in ascending order. 3.3.1.2. Else if the user enters '2', the delivery status of the parcel will be sorted in descending order. 3.3.1.3. Else the system will return the user to the main page. 3.3.1.4. End if 3.3.2. Function displayBubbleSortMenu will be executed with option, parcel and count as the arguments. 3.3.3. The system displays the delivery status of the parcels based on the user selected choice using bubble sort. 3.4. If the user chooses case '4', the user will prompt the system to sort the parcels' shipping options. 3.4.1. The system will give the user two options: bulky and heavy delivery or standard delivery. 3.4.2. The system will give the user two options: either to sort in ascending or descending order. 3.4.2.1. If the user enters 'A', the shipping category of the parcel will be sorted in ascending order. 3.4.2.2. Else if the user enters 'B', the shipping category of the parcel will be sorted in descending order. 3.4.2.3. Else the system will return the user to the main page. 3.4.2.4. End if 3.4.3. Function displaySelectionSortMenu will be executed with category, parcel and count as the arguments. 3.4.4. The system displays the parcels' shipping category based on the user selected choice using selection sort. 3.5. If the user chooses case '5', the user will prompt the system to sort the parcels' tracking number. 3.5.1. The system will give the user two options: either to sort in ascending or descending order. 3.5.1.1. If the user enters '1', the parcel's tracking number will be sorted in ascending order. 3.5.1.2. Else if the user enters '2', the parcel's tracking number will be sorted in descending order. 3.5.1.3. 3.5.1.4. Else the system will return the user to the main page. End if 3.5.2. Function displayMergeSortMenu will be executed with option, parcel and count as the arguments. 3.5.3. The system displays the parcel's tracking number based on 4. the user selected choice using merge sort. After the sorting has been done, the system will display the parcel's 5. tracking number, address, sender name, receiver name, parcel's delivery status (complete or incomplete), or parcel's shipping option. After that, the system will prompt the user either to continue or quit the 6. system. 5.1. If the user selects '0', the system will terminate. 5.2. Else the system will display a message to the user to try again. End 4. Description of how to implement data structure operations: sorting and searching. In this program, we apply a variety of types of sorting and searching. For searching a parcel, we apply binary search to our function. We use binary search because of its efficiency for searching in sorted arrays or lists. Also, it has a time complexity of O(log n), which makes it significantly faster than linear search for large datasets. Not to mention, the algorithm is quite simple and easy to understand. For sorting a parcel using its delivery status by ascending or descending order, we utilize bubble sort to our function. It has simple sorting algorithms making it easy to understand, not to mention its adaptability with partially sorted data. For sorting a parcel using its shipping category by ascending or descending order, we use selection sort sort to our function. It also has simple sorting algorithms making it easy to understand, not to mention its in-place sorting, meaning it does not need additional memory space. For sorting a parcel using its tracking number by ascending or descending order, we use merge sort sort to our function. It has a stable sorting algorithm, meaning that equal elements maintain their relative order in the sorted output. Also, it performs consistently regardless of the initial order of the elements. Then, merge sort is good for its efficiency for linked lists, as it can easily be adapted to work with linked data structures.