Benha University

Benha Faculty of Engineering

Data Structure and Algorithm Research Topics

3rd Electric

Communication and Computer

2019/ 2020

Contents

Policy

Dear students, in order to complete your research choose only one topic from the mentioned topics in this document. You should implement **several** different data structures and algorithms, and **compare** their performance through rigorous experimental tests. The end result should be documented in a report giving a description of the techniques implemented and tested, the tests performed and the experimental results. You must cover all the mentioned requirement in order to pass your evaluation

Implementation report should be of high quality, i.e., it should be well organized, formatted and contain valuable information.

Any student who is caught submitting work that they did not do themselves may receive a NOT ACCEPTABLE research.

Any group of two or more students handing in sufficiently similar research will all receive NOT ACCEPTABLE research.

Plagiarism and cheating are viewed as being particularly serious. Plagiarism and cheating, i.e. presenting another's ideas, words, program, algorithm or images as your own, using unauthorized co-operation or collaboration or completing work for another student — is NOT ACCEPTED.

Topic 1

Towers of Hanoi Problem

Description

The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape. The objective of the puzzle is to move the entire stack to another rod, obeying the following rules:

1. Only one disk must be moved at a time.
2. No disk may be placed on top of a smaller disk.

Basic Requirement

You are required to implement this puzzle by different data structures (at least two data structure) and make comparison between the results of each method. Explain the algorithm you used. you can use a new technique you didn't study but you must include explanation of it in the report.

Find the minimal number of moves required to solve a Tower of Hanoi puzzle.

Report

Write a report with your work, the report must include (at least):

1- Introduction
2- Problem solving Steps
   Include C++ program code for each step, in case of using any new technique explain it.

3- implementation Result
   Include the comparison, time to run any program
4- Conclusion
   Include the best technique choice from your point of view.
5- References

Note: Format your document properly, don’t forget to make a cover page includes the topic title, your name and your section.

Topic 2

Knapsack Problem

Description

suppose you are planning a picnic. You've constructed a list of items you would like to carry with you on the picnic. Each item has a weight associated with it and your knapsack is limited to carrying no more than 15 pounds. You have also come up with a 1 to 10 rating for each item, which indicates how strongly you want to include the particular item in the knapsack for the picnic. Write Program to select your items.

Item	Weight	Rating
Ant Repellent	1	2
Blanket	4	3
Bratwurst	3	8
Brownies	3	10
Frisbee	1	6
Salad	5	4
Watermelon	10	10

Basic Requirement

You are required to use different data structures and algorithm techniques and make comparison between the results of each method. Below are the minimum techniques you can use (you are allowed to use more if you want)

1- Write your program using Stack data structure
2- Write your program using Queue data Structure
3- Using an array data structure, apply two different algorithm techniques to find the solution. (you can use any algorithm technique you have studied, optionally you can use a new technique you didn't study but you must include explanation of it in the report)

Report

Write a report with your work, the report must include (at least):

1- Introduction
2- Experiment Steps

Include the program code for each step, in case of using any new technique explain it.

3- Experimental Result
   Include the selected items, time to run any program
4- Conclusion
   Include the best choice from your point of view.
5- References

Note: Format your document properly, don't forget to make a cove page includes the topic title, your name and your section.

Topic 3

Bridge Crossing at Night

Description

Four people need to cross a rickety footbridge; they all begin on the same side. It is dark, and they have one flashlight. A maximum of two people can cross the bridge at one time. Any party that crosses, either one or two people, must have the flashlight with them. The flashlight must be walked back and forth; it cannot be thrown. Each person cross the bridge in different time:

Person	Time
1	1 min
2	2 min
3	5 min
4	10 min

For example, if person 4 returns the flashlight, a total of 20 minutes have passed. Calculate the total time they need to cross the bridge?

Basic Requirement

You are required to use different data structures and algorithm techniques and make comparison between the results of each method. Below are the minimum techniques you can use (you are allowed to use more if you want)

1- Write your program using Stack data structure
2- Write your program using Queue data Structure
3- Using an array data structure, apply two different algorithm techniques to find the solution. (you can use any algorithm technique you have studied, optionally you can use a new technique you didn't study but you must include explanation of it in the report)

Report

Write a report with your work, the report must include (at least):

1- Introduction
2- Experiment Steps
   Include the program code for each step, in case of using any new technique explain it.
3- Experimental Result

Include the calculated time in each case to cross the bridge, time to run any program

4- Conclusion
   Include the best choice from your point of view.
5- References

Note: Format your document properly, don’t forget to make a cover page includes the topic title, your name and your section.

Topic 4

Student Management System using Linked list and Binary Search Tree BST

Description

In this project, the students implement a student database system using dynamic memory allocation for linked list and binary tree. It is a complete system with the functionality of add record, search record, Update record and delete record using linked list. If students understand basic CRUD (Create, Read, Update and Delete) functionality it would be easy for them to create any Management System using this functionality.

The menu is handled by if else statement. While adding a new record, if any node already exist then new node is connected with it otherwise a new node is created. While searching a node, a pointer searches all the pointers of linked lists or BST to match the required data from start to end. While Modification the data part is overwritten with new data and in case of Deletion, the pointers before and after of the node which we need to delete are connected with each other.

Basic Requirement

Implement this system by linked list and BST and compare the results. You will use:

1. Pointers
2. Loops
3. Functions
4. Linked List, BST
5. If Else

and the major function in menu

- Add New record
- Search a record
- Modify a record
- Delete a record
- Traverse all record including (DFS and BFS) in BST.
- Exit

Report

Write a report with your work, the report must include (at least):

1- Introduction
2- Experiment Steps
   Include C++ program code for each step, in case of using any new technique explain it.
3- Experimental Result
   Include the calculated time in each case to cross the bridge, time to run any program
4- Conclusion
   Include the best choice from your point of view
5- References.

Note: Format your document properly, don't forget to make a cove page includes the topic title, your name and your section.

Topic 5

Data Structure & Algorithm Techniques Comparisons

Description

You have been studied some data structure and algorithms techniques, it is needed to write a menu driven program to implement some of them, and compare the results.

Basic Requirements

1- Data Structure
   Stack, queue, linked list and Binary search tree
2- Algorithms
   At least 4 sorting techniques, and 2 Searching Techniques

Report

Write a report with your work, the report must include (at least):

1- Introduction
2- Experiment Steps
   Include the program code for each step, in case of using any new technique describe it.
3- Experimental Result
   Include the time to run any program
4- Conclusion
   Include the best choice from your point of view.
5- References

Note: Format your document properly, don't forget to make a cove page includes the topic title, your name and your section.

Topic 6

Sorted Singly Linked Lists

Description

In this project, we store a finite list of n linearly ordered elements in that order stored in singly linked nodes. If there are zero elements in the list, the list is said to be empty. Each element is stored in an instance of the Single_node. If the list is empty, the head and tail pointers are assigned nullptr. Otherwise, the head pointer points to the first node, the tail pointer points to the nth node, the next pointer of the ith node ($1 \leq i < n$) points to the (i+ 1) st node, and the next pointer of the nth is assigned nullptr.

Basic Requirements

For this project, you will implement at least two efficient algorithms to sort singly linked linear lists of integers. We'll write something like [5, 8, 4, 9, 1, 2, 3, 7, 6] to mean such a list. This list has nine nodes. The integers in the list don't decrease as we visit them from left to right. For example, if we sort [2, 3, 1, 1, 2] into nondecreasing order, then we get [1, 1, 2, 2, 3]. Similarly, if we sort [5, 8, 4, 9, 1, 2, 3, 7, 6] into nondecreasing order, then we get [1, 2, 3, 4, 5, 6, 7, 8, 9]. We'll use this list as a running example in the rest of this description.

- We also want our sorting algorithm to be as efficient as possible. Use at least 2 sorting techniques and compare the result.
- If we need to insert a value to after sorted the linked list, write a function to insert the value in a sorted way.
- For the two algorithms you used, implement them to sort array with the same size of the linked list and compare the result of sorting an array and sorting a linked list, you may use large n enough to show the difference.

Accessors

- void insert (); Creates a new Single_node storing the argument, placing it into the correct location in the linked list such that the element in the previous node (if any) is less than or equal to the element stored in the current node, and that element is less than or equal to the element stored in the next node. The head and tail pointers may have be updated if the new node is placed at the head or tail of the linked list.
- sortList() will sort the nodes of the list in ascending order.
- display() Traverse through the list till current points to null.
- bool empty(); Returns true if the list is empty, false otherwise.
- Type front(); Retrieves the object stored in the node pointed to by the head pointer. This function throws a underflow if the list is empty.
- Type back();Retrieves the object stored in the node pointed to by the tail pointer. This function throws a underflow if the list is empty.
- Type pop_front(); Delete the node at the front of the linked list and, as necessary, update the head and tail pointers. Return the object stored in the node being popped. Throw an underflow exception if the list is empty
- int erase(); Delete the first node (from the front) in the linked list that contains the object equal to the argument (use ==) to test for equality with the retrieved element). As necessary, update the head and tail pointers and the next pointer of any other node within the list. Return the number of nodes that were deleted.

Report

Write a report with your work, the report must include (at least):

1- Introduction
2- Experiment Steps
   Include the program code for each step, in case of using any new technique explain it.
3- Experimental Result
   Include the time to run any program
4- Conclusion
   Include the best choice from your point of view.

5- References

Note: Format your document properly, don’t forget to make a cover page includes the topic title, your name and your section.