National University

of Computer & Emerging Sciences Peshawar Campus

Student Name: _____ Roll No: _____

Program: BS(CS)
Semester: Spring-2020
Time Allowed: 03 hours
Course: CS218 – Data Structures

Examination: Final
Total Marks: 70 Weightage: 50
Date: 22/06/2020
Instructor Name: Waqas Ali

NOTE: Attempt all questions.

1. Consider a list of persons (names) and their phone numbers. You have to create a contact list (names and the phone numbers). You do not need to add more contacts to the list but frequently search for contacts based on their names in the list.
   a. Among the list data structures you have studied, recommend one data structure that is suitable for this scenario also state your arguments to justify your selection.
   b. Provide definition of the class(es) required for its implementation. You do not need to implement the functions such as insert( ), delete( ), search( ) etc. Just list the class(es) and their data members.

   (5 + 5 = 10 Points)

2. Consider the same contact list as in **Question 1** but with an added requirement that extra contacts needs to be added to the list regularly. Searching is still a priority and is performed on the names of the contacts.
   a. Suggest any data structure/process apart from the list based data structures for this scenario. Provide arguments to justify your selection.
   b. Using a contact list as an example show how (diagrammatically) would you insert the contacts in to the data structure you have chosen.

   (5 + 5 = 10 Points)

3. In a Queue data structure, we have two ends: front (for removal) and rear (for insertion). In array based implemention of Queues, enqueue (insertion) cannot be performed when the rear is at the last index of the array even if there are slots available at the start of the array, example below:

		5	2	6	8	9	12
0	1	2	3	4	5	6	7
		front					rear

One way to solve this problem is to picture the array as a circular array i.e. allow the rear and front indexes jump (wrap around) from the end to the start of the array. Consider if the size of the queue (number of elements) is kept fixed. Can the same problem explained above be solved using a *Circular Linked Linked*

list? If yes, then provide the queue implementation (in C++) considering you already have the circular linked list class. If you think the problem cannot be solved using a circular linked list then, state your arguments to prove your point. **(10 Points)**

4. A barber wants to maintain record of their clients. Each client is reprented by a client id, their name, the barber would also like to keep a record of the client's gender.
    a. Suggest a data structure which is suitable for this scenario, keeping in mind the following operations to be performed on the data (justify your selection with arguments):
        i. Insert(client_id, name, gender) – Insert a new client.
        ii. Update(client_id, name) – Update the name of the client specified by the client id.
        iii. ClientDiff(client_id) – Difference between the number of men and women with the id less than client_id
    b. Design and implement the class(es) requied for this scenario including the opertions mentioned above. You do not need to write code of the data structure you choose for this scenario (for example if you choose linked list you do not need to implement its functions like insert etc. If you need them just call them. Consider them to be provided to you)

(5 + 10 = 15 Points)

5. Consider the following list of tasks and their priorities.

Tasks	Priority
Task 1	2
Task 2	4
Task 3	1
Task 4	5
Task 5	3
Task 6	1

Create a class named **PriorityTasks** with two fileds – One for the name of the task and the other one for the priority. Each task is to be performed based on its priority (higher number needs to be performed first).

    a. Suggest a data structure such that the task with the highest priority can be accessed easily (i.e. with the lowest number of operations). You are not allowed to sort the list. Justify your selection with arguments.
    b. Based on your selection of the data structure show how each of the task will be inserted into the data structures. Show each step diagrammatically.

(5 + 5 = 10 Points)

6. Encode the following phrase using Huffman Codes and show each step of the process: **(5 Points)**

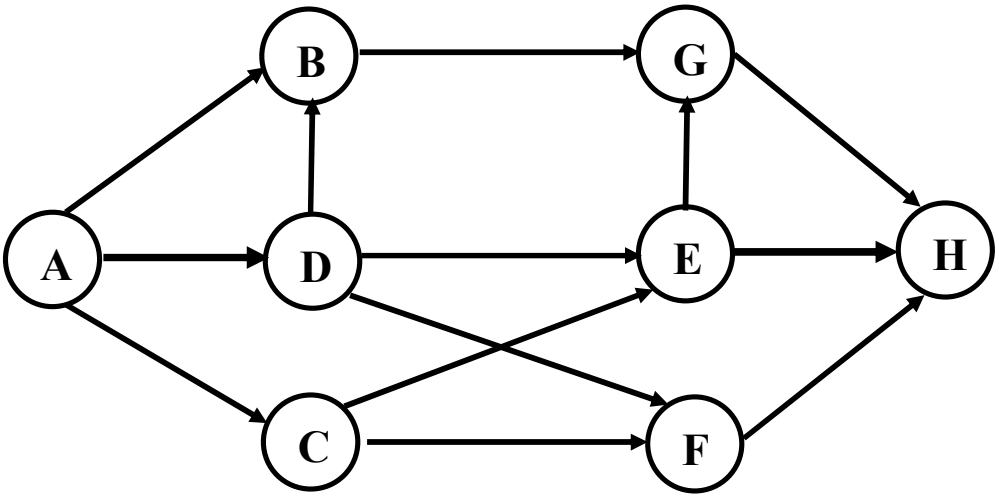
The best of both worlds

7. Construct an expression tree of the following postfix expression and show each step of the process:

4 2 + 3 5 1 - * +

(5 Points)

8. Consider the following graph:



a. Provide adjacency list representation of the graph.

b. Provide the breadth first traversal of the graph.

c. Provide the breadth first traversal of the graph.

(1 + 2 + 2 = 5 Points)

The End