M. Tech -Data Science and Engineering
I Semester 2018-19 (Oct 2018 to Mar 2019)
Mid-Semester Examination

Course No	:	DSEABZG519/DSEADZG519/ DSEAHZG519
Course Name	:	Data Structure and Algorithm Design
Nature of Exam	:	Closed Book
Max Marks	:	30
Duration	:	1-Hour & 30-Minutes
Date of Exam	:	23rd December 2018

No. of page: 2
No. of questions: 4

Session: FN.

Note:

1. Please read and follow all the instructions given on the cover page of the answer booklet.
2. All questions have part a and part b.
3. Start each answer from a fresh page. All parts of a question should be answered consecutively.
4. Please ensure that your answers cover necessary technical details, avoiding unnecessary text and diagrams.

---

1) **a.** You are planning the seating arrangement for an alumni meet, given a list of people, P. Suppose you are also given a lookup table **L**, where $\boldsymbol{L[u]}$ for $\boldsymbol{u \in P}$, is a list of guests that ***u*** is ***a friend of***. If ***u*** is *a friend of* **v**, then ***v*** is *a friend of* ***u***. You are required, to arrange the seating such that, any guest at a table is *a friend of* every other guest sitting at the same table either directly or through some other guests sitting at the same table. For example, if *u* is *a friend of v*, and *v* is *a friend of w*, then *u*, *v*, *w* can sit at the same table.

i. Describe an efficient algorithm that, given ***P*** and ***L***, returns the **minimum** number of tables needed to achieve this requirement. **3.5M**

ii. Give the running time of your algorithm. **1.5M**

b. The array-based **Queue** throws an exception, when the array’s capacity has been reached. Consider one of the following alternatives in which we use a resize method to expand the size of the array, each time we run out of array space. The cost of a resize that makes the array larger is proportional to the new size.

i. Suppose, we expand the array’s capacity by one element. Give an analysis of the running time (Big-Oh notation) in the worst-case, for a sequence of N insertions? **1.5M**

ii. Suppose, we double the array’s capacity (assume the capacity is not zero). Give an analysis of the running time (Big-Oh notation) in the worst-case, for a sequence of N insertions? **1M**

2) **a.** We have learned several different ways of implementing a collection of objects. Consider only the following collections in answering these questions. The collections are **5M**

1. Sorted list implemented with an array
2. Unsorted list implemented with an array
3. Heap implementation of a priority queue
4. Hash table
5. Vector implemented with an array

Find the **worst case time** to perform the following operations in each of the collections (numbered i to v listed above). Your answer can be one of the following

$$\mathbf{O(1) / O(\log n) / O(n) / O(n \log n) / O(n^2) / O(n^3) / O(2^n)}$$

- **isPresent:** return the element depending on whether a particular object is a member of a collection
- **insert**: add a new item to the collection.

b. Suppose you are given an array A[1 : n] stored in read-only memory from which you want to sample ***k*** elements **uniformly** at random *without replacement* (so all of the sampled elements are distinct). Show how to do this, in O(n) expected time and O(k) space using an ADT covered in the contact sessions, and **do not** assume that the elements of ***A*** are *integer-valued*. **2.5M**

3) **a.** Write an algorithm using **Stack** Data structure to test if a mathematical expression has **balanced and matching parentheses**. This algorithm will accept an expression character by character as input along with a stack. The expression can use one of the following parenthesis : [,], (,), {,} **5M**

The input expressions can also have variables and operators, which must be ignored from processing. For example, an expression [a + {c * d} (e+d)] has matching parenthesis leaving out the variables and operators. That is, the parenthesis sequence [{ } ( )], ignoring the variables and operators, is a valid sequence. Further, the expression [a + (c * d } {e+d)] do not have matching parenthesis as in the corresponding parenthesis sequence [(} {)], the curly braces does not match.

b. Construct the binary tree, whose post order traversal: B D F E O L K N P G and In order traversal: B E D F G K O L P N. **2.5M**

1. Display the pre order traversal
2. What is the time complexity of performing the preorder traversal on the tree?

4) **a.** Given a **hash table** of size **7** and hash function ***h(x) =x mod 7***, show the final table after inserting the following elements in the table ***19, 26, 13, 48, 17*** for each of the cases **2.5M**

1. When ***linear probing*** is used
2. When ***double hashing*** is used with a second function **g(x) = 5 – (x mod 5)**

b. i. Write an algorithm to find the **k**$^{\text{th}}$ **largest** element of an array using a

- ○ **Min** Heap. **2M**
- ○ **Max** Heap **2M**

ii. What is the time complexity in both cases if the array is **unsorted**? **1M**