BPDC, Dubai - Second Semester, 2018-2019

Course No: CS F211 | **TEST1 (Closed Book)** | **Course Title: DSA**
Date: 24th FEB 2019 | | **Weightage: 20%**
Duration: 8.30 – 9.20 am | **Computer Science** | **Max. Marks. 40**

Note: Show all calculations/steps clearly.

1. Evaluate the postfix expression using stack. Show your work step by step clearly in the format given below:
23*21-/53*+

Input read	Stack contents	Action & Reason

[5 Marks]

2. What is the time complexity for the following operations? Express using big-oh notation. Justify your answer in a line or two.

a. Insertion into a stack
b. Insertion into a linked list at an **index > 0**. Note index=0 is the beginning of the list.
c. Finding an element in the list
d. Enqueue an element in the queue
e. Matching parenthesis using stack. Suppose the input has n parenthesis.
[5 * 1 = 5 Marks]

3. Suppose we have a list of fruits. It is required to represent it sequentially as linked list in the order given in the following table. Draw the diagram illustrating the list in the following cases:

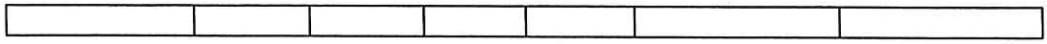
a. Singly linked list
b. Doubly linked list
c. Circular singly linked list
d. Circular doubly linked list
Clearly indicate the index of the nodes, start node, value in the data and pointer fields of each node.

Node name	Data field	Address of the node
Node1	Apple	1000
Node2	Mango	2000
Node3	Grapes	3000
Node4	Pear	4000

e. In general when do we have error conditions when we delete an element from a singly linked list?
[5*1 = 5 Marks]

4. Consider an array based implementation of STACK. Assume that the array's size is N = 4 elements maximum. Show the contents of the STACK (trace through) at each step, for the following sequence of ten operations. Make sure to denote exceptions like empty/full. Also give the value of top after each operation is performed. Initially top is -1 for an empty stack.
Operations: PUSH Z, PUSH X, PUSH F, PUSH D, PUSH B, POP, POP, POP, POP, POP

Operation	A[0]	A[1]	A[2]	A[3]	Exception Condition (if any)	Top



Algorithm *push*(o)	**Algorithm *pop*()**	**Algorithm *size*()**
if *size()* = *N* then *Full Stack Error* else *top* ← *top* + 1 *S[t]* ← *o*	if *isEmpty*() then *Empty Stack Error* else *e* ← *S[top]* *S[top]* ← *null* *top* ← *top* – 1 return *e*	return *(top* + 1) **Algorithm *isEmpty*()** return *(top* = -1)

[10 *0.5 = 5 Marks]

5. With proper justification express the time complexity of the following for loops using Big-oh notation

a.

```
void function(int n)
{
    int count = 0;
    for (int i=1; i<=n*n; i++)
        for (int j=0; j<i; j++)
            count++;
}
```

b.

```
void function(int n)
{
    int count = 0;
    for (int i= 1; i<=n; i++)
        for (int j=1; j+n/2<=n; j = j++)
            for (int k=1; k<=n; k = k * 2)
                count++;
}
```

[2 * 3 =6 Marks]

6. Write the pseudocode for recursive factorial computation for a given n. Represent the execution time using a recurrence equation and solve it using iterative substitution method.
[2+2+4 = 8 Marks]

7. Prove with proper steps which of the following is not $O(n^2)$?

a. $n^{1.98}$
b. n^3 / (sqrt(n))
c. $10n^2+20$
[2 * 3 = 6 Marks]

BPDC, Dubai - Second Semester, 2018-2019

Course No: CS F211	**TEST2 (OPEN BOOK)**	**Course Title: DSA**
Date: 08th APR 2019		**Weightage: 20%**
Duration: 8.30 – 9.20 am	**Computer Science**	**Max. Marks. 40**

Note: Show all calculations/steps clearly.

1. Consider a Bloom filter with m=5 and k = 2. Let the two hash functions be $h_1(k) = k \, mod \, 5$ and $h_2(k) = (k+3) \, mod \, 5$.
   a. Show the initial state of the bloom filter.
   b. Show the state of the bloom filter after Insert (9)
   c. What is the answer for Query (30)? Is it a false positive? Why or Why not?

[1 + 2 + 2 Marks]

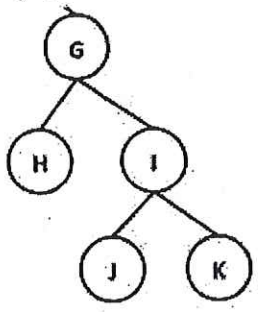
2. a) Suppose the numbers 12, 44, 52, 58, 38, 27 are inserted in that order into an initially empty AVL tree. Draw the AVL tree after each insertion. Clearly show the heights of the nodes and the imbalances if any after every insertion. Perform necessary rotations to make it an AVL tree. Mention the rotations done.

b) Delete node with key 44 from the resulting tree. Clearly show the heights of all the nodes in the resulting tree after deletion. Perform rotation if needed to convert the resulting tree into an AVL tree.

Note: During deletion of a node with two children always replace the deleted node with the key from its inorder successor. While rotations choose x such that you perform single rotation.

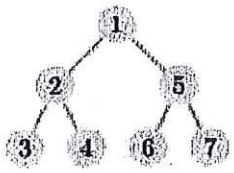
[8+2 marks]

3. a) Consider the following binary tree. Is the given tree a min-heap? Justify.



b. Consider the following min-heap. Perform one delete() operation on the min heap. Show the resulting min-heap after the delete operation. Show your work clearly.

PTO

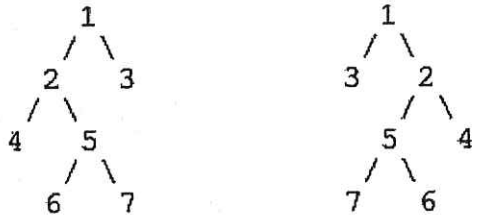


[2 + 3 marks]

4. Write a pseudocode named mirror() that takes a reference to the root node of a binary tree and creates a new tree (with its own nodes) that is the mirror image of the original tree.

For example: If root is a reference to the root of the tree on the left below, then the return value of mirror(root) would be a reference to the root of the tree on the right below.

Hint: This method is much easier to write if you use recursion.



[7 marks]

5. Arrange nodes that contain the letters: A, C, E, L, F, V and Z into two binary search trees:

a). one that has max height

b). one that has min height.

.[3+3 marks]

6. Given Preorder, Inorder, Postorder Traverals of a Binary Search tree; Name two traversals which are sufficient to reconstruct the original tree. Name two traversals which are insufficient to reconstruct the original tree. Justify your answers with proof.

[7 marks]

BITS PILANI DUBAI CAMPUS, DIAC, DUBAI
CS F211 DATA STRUCTURES AND ALGORITHMS
SECOND SEMESTER 2018 - 2019
COMPREHENSIVE EXAMINATION
CLOSED BOOK

Date & Session : 19 – MAY – 2019 & FN **Time: 8.30 am – 11.30 am**
Weightage : 35% (70 Marks)
Total No. of pages: 5 **Total number of questions: 6 + 6**

Note: Answer PART A and PART B in separate answer books.

PART A (35 MARKS)

1. Consider the undirected graph given below in *Figure1*

a) Give the order in which the vertices are visited for the graph by the BFS traversal starting at node A
b) Give the order in which the vertices are visited for the graph by the DFS traversal starting at node A
c) Give the adjacency matrix representation of the graph
d) Give the adjacency list representation of the graph

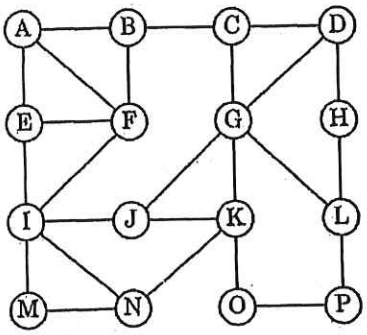


Figure 1

[4 * 2 = 8 M]

2. Answer the following questions with either true or false. Assume there are n elements in the datastructure. Justify your answer.

a) One can implement a queue based on a linked list so that EVERY INDIVIDUAL enqueue operation is time $O(1)$.
b) The core datastructure of Breadth-First Search is a queue.
c) One can check whether a given item is contained in a linked list of length n in time $O(\log n)$.
d) Removing an Integer from a singly-linked list of Integer can be done in worst-case time $O(n)$.

e) Removing an Integer from a doubly-linked list of Integer can be done in worst-case time $O(1)$.
f) A binary tree of height h cannot contain more than h^2 leaves.
g) Inserting an item into an unbalanced binary search tree with n elements may require time $O(n)$.

[7 * 1 = 7M]

3. Insert the following keys into an initially empty binary search tree 6,4,8,1,12,3,19,5,20. Draw the tree after each insertion.

[4 M]

4. For the following graph in *Figure 2* the bold edges form a Minimum Spanning Tree. What can you tell about the range of values for x? Justify your answer.

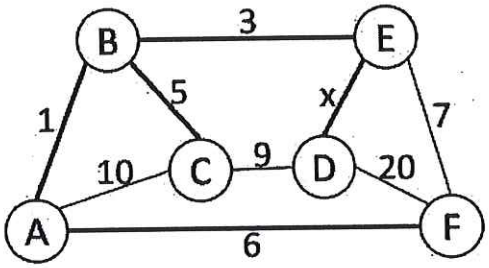


Figure 2

[2 M]

5. a) Assume a Bloom filter is used to Preventing Weak Password Choices. A dictionary of easily guessable passwords is stored as a bloom filter. The Bloom filter is queried whenever users pick passwords. Assume new entries can be added to the Bloom filter. What is a false positive in this context?

b) You are given a binary min heap of height h. What is the minimum and maximum number of comparisons we might have to do when inserting a value (in terms of h)?

c) What is the worst case running time for **Delete** in a Binary Search Tree of size N. Assume that the most time-efficient implementation is used. Assume that all keys are distinct. Justify your answer

d) What is the postfix form of the expression represented by the tree in *Figure 3*?

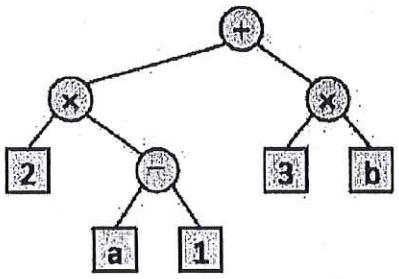


Figure 3

e) Consider the following tree in *Figure 4.* Is it an AVL Tree? Justify your answer.

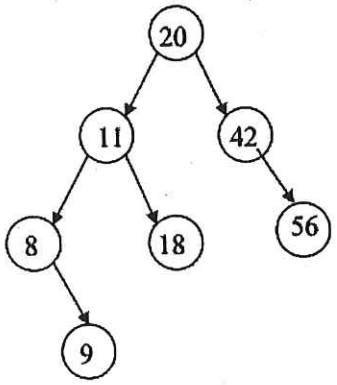


Figure 4

[1 + 1 + 1 + 1 + 2 M]

6. Consider the graph shown in *Figure 5.* Compute the Minimum Spanning Tree (MST) from the source node labelled A using **Prim's** algorithm. Show your work step by step using a table as shown below with fields like vertices, known, distance (d_v) and previous node (p_v). Show the computations at each step clearly. Draw the MST and find the cost of the MST.

vertex	known	d_v	p_v

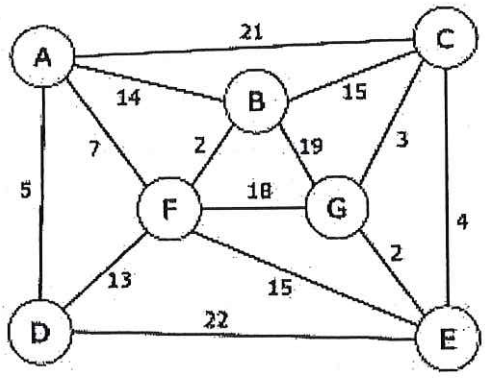


Figure 5

[5 + 2 + 1 M]

****

PART B (35 MARKS)

1. Find first k maximum occurring words in a given set of strings? Which data structure would be best choice for this problem? Justify your answer. Also write a pseudocode which uses that data structure to most efficiently solve the said problem.

Input:

```
keys = ["code", "coder", "coding", "codable", "codec", "codecs", "coded",
        "codeless", "codec", "codecs", "codependence", "codex", "codify",
        "codependents", "codes", "code", "coder", "codesign", "codec",
        "codeveloper", "codrive", "codec", "codecs", "codiscovered"]

k = 4
```

Output:

```
codec occurs 4 times
codecs occurs 3 times
code occurs 2 times
coder occurs 2 times
```

[6 M]

2. Show the B-tree of order 5 that results when inserting the elements R, Y, F, X, A, M, C, D, E, T, H, V, L, W, G (in that order) into an initially empty B-tree. You need only draw the trees just before and after each split.

[6 M]

3. a) Consider a hash table using separate chaining for storing integer values in the range from 0 to 9999. Consider a table size (size of the array) of 1000. What is the maximum size that one of the linked lists can reach? Justify mathematically.

b) What does a hash function do? What are two desirable properties of a hash function?

[3 + 2 M]

4. a) Let $f(n) = kn^3$ where k is a constant, and let $g(n) = n^3-2$. Claim: $f(n) = O(g(n))$. Justify through mathematical proof, whether this claim is true or false.

b) There are four algorithms A1, A2, A3 and A4 to solve a given problem with the order log(n), log(log(n)), n log(n), n/log(n) respectively. Sort the algorithms according to the time complexity beginning from the fastest to the slowest. Which is the best algorithm?

[3 + 3 M]

5. a) Under what circumstances will it be faster to use Insertion Sort to sort an array than to use QuickSort?

b) Explain what it means for a sorting algorithm to be stable, and give one example of a stable sorting algorithm and one example of an unstable sorting algorithm.

c) Can every recursive function be converted into equivalent iterative function? Justify your answer.

[3*2 = 6 M]

6. a) Given the following array [10, 5, 3, 9, 22, 24, 28, 27, ?] and assuming that Quicksort will be used to sort this array in ascending order, select a value for the last element of the array (indicated by "?") such that the partitioning performed by Quicksort is most balanced. Explain why this makes Quicksort perform efficiently.

b) Show the results of the first two rounds of the Quicksort algorithms based on the number you have chosen.

[2 + 4 M]

****

BPDC, Dubai - Second Semester, 2018-2019

Course No: CS F211	**QUIZ 1 (Closed Book)**	**Course Title: DSA**
Date: 14th MAR 2019		**Weightage: 10%**
Duration: 8.45 – 9.05 am	**Computer Science**	**Max. Marks. 10**

Note: Show all calculations/steps clearly.

1. Draw the 7-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94 in this order using the hash function $h(k) = (k) mod\ 7$ and assuming collisions are handled by quadratic probing. If overflow occurs, say so, and indicate the element that causes the overflow. Compute the load factor of this hash table if there are four elements in it. **[3 +1 M]**

2. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: 25, 15, 11, 40, 90, 120, 80, 62. Guess the pivot used for partitioning. Justify your answer. **[1 M]**

3. Suppose the partition algorithm of quick sort always produced a 8-to-2 split. Show by recursion tree method that the time complexity of Quick sort for this split is O (n log n). **[2 M]**

4. Draw the execution of merge sort as a binary tree illustrating the working of merge sort for the following input numbers A = 38, 27, 43, 3, 9, 82, 10. Also indicate the various calls to merge and the mergesort algorithms for this input sequence clearly indicating the values of the parameters for each call. **[3 M]**

BPDC, Dubai - Second Semester, 2018-2019

Course No: CS F211	**QUIZ 1 SCHEME**	**Course Title: DSA**
Date: 14th MAR 2019		**Weightage: 10%**
Duration: 8.45 – 9.05 am	**Computer Science**	**Max. Marks. 10**

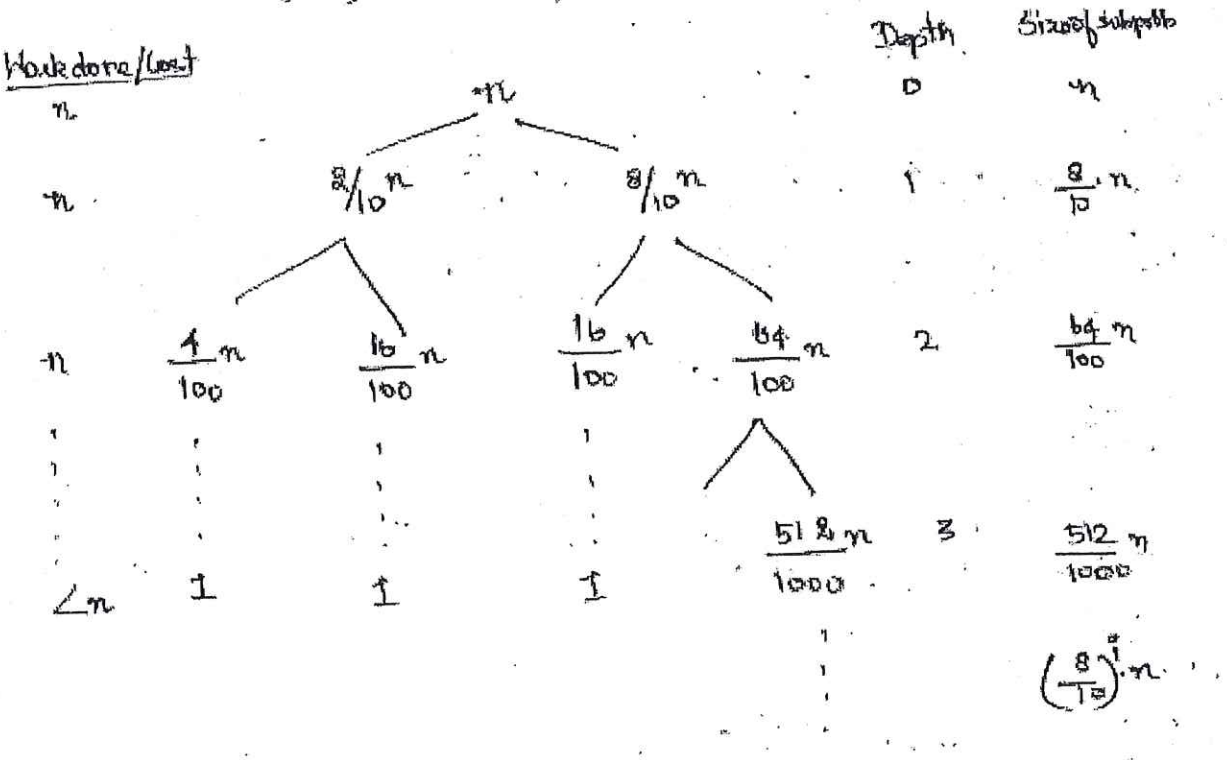
Note: Show all calculations/steps clearly.

1. Draw the 7-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94 in this order using the hash function $h(k) = (k) mod\ 7$ and assuming collisions are handled by quadratic probing. If overflow occurs, say so, and indicate the element that causes the overflow. Compute the load factor of this hash table if there are four elements in it. **[3 +1 M]**

2. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: 25, 15, 11, 40, 90, 120, 80, 62. Guess the pivot used for partitioning. Justify your answer. **[1 M]**

3. Suppose the partition algorithm of quick sort always produced a 8-to-2 split. Show by recursion tree method that the time complexity of Quick sort for this split is O (n log n). **[2 M]**

$$T(n) = T\left(\frac{2}{10}\cdot n\right) + T\left(\frac{8}{10}\cdot n\right) + n$$



No. of levels $= \left(\frac{8}{10}\right)^i n = 1 \Rightarrow n = (10/8)^i$

Assume n to be a perfect power of $(10/8)^i$

$\Rightarrow \log_{10/8} n = i \rightarrow \log n$.

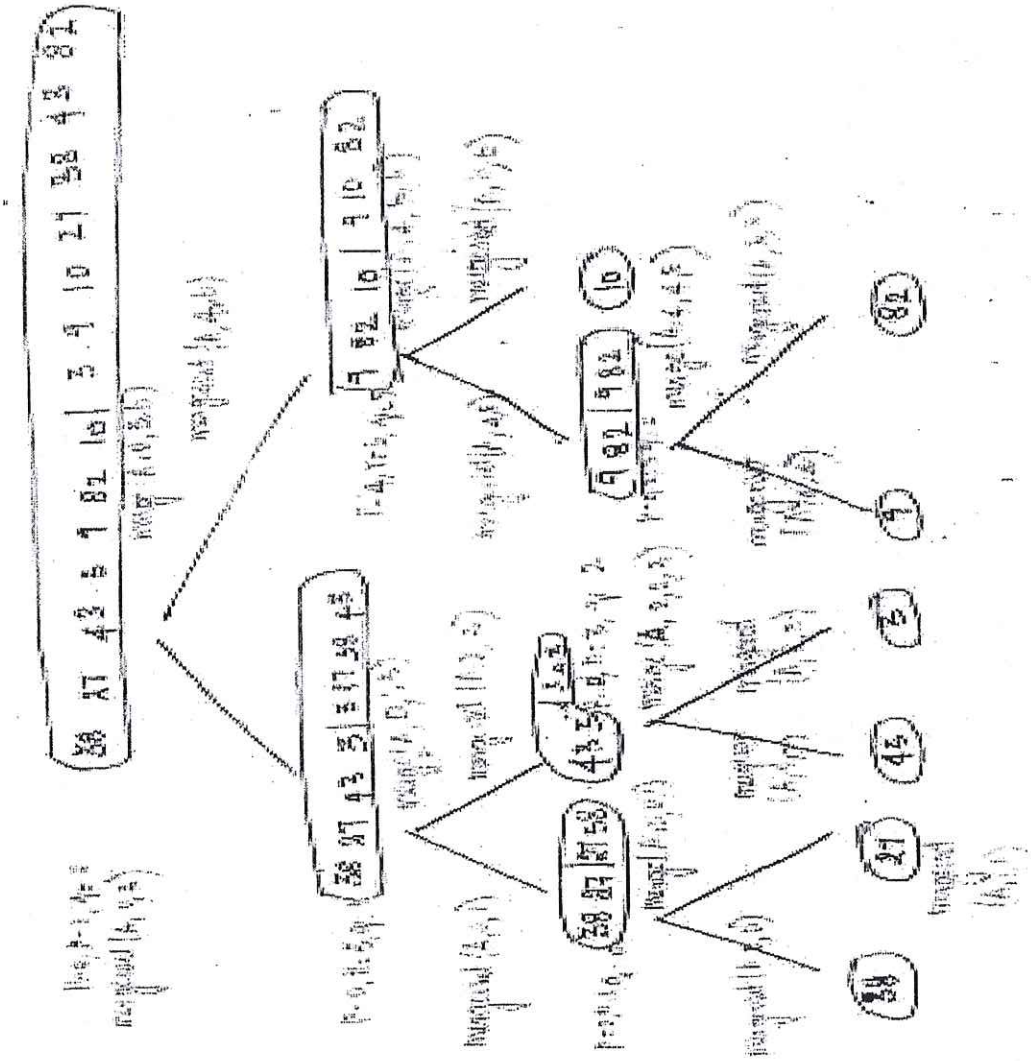
∴ Total Cost = Cost @ each level * (no. of levels + 1)

$= n * (\log n + 1)$

$= n \log n + n$

$= O(n \log n)$

4. Draw the execution of merge sort as a binary tree illustrating the working of merge sort for the following input numbers A = 38, 27, 43, 3, 9, 82, 10. Also indicate the various calls to merge and the mergesort algorithms for this input sequence clearly indicating the values of the parameters for each call. **[3 M]**



BPDC, Dubai - Second Semester, 2018-2019

Course No: CS F211	**QUIZ 1 (Closed Book)**	**Course Title: DSA**
Date: 14th MAR 2019		**Weightage: 10%**
Duration: 8.45 – 9.05 am	**Computer Science**	**Max. Marks. 10**

Note: Show all calculations/steps clearly.

1. Draw the 7-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94 in this order using the hash function $h(k) = (k) mod\ 7$ and assuming collisions are handled by quadratic probing. If overflow occurs, say so, and indicate the element that causes the overflow. Compute the load factor of this hash table if there are four elements in it. **[3 +1 M]**

2. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: 25, 15, 11, 40, 90, 120, 80, 62. Guess the pivot used for partitioning. Justify your answer. **[1 M]**

3. Suppose the partition algorithm of quick sort always produced a 8-to-2 split. Show by recursion tree method that the time complexity of Quick sort for this split is O (n log n). **[2 M]**

4. Draw the execution of merge sort as a binary tree illustrating the working of merge sort for the following input numbers A = 38, 27, 43, 3, 9, 82, 10. Also indicate the various calls to merge and the mergesort algorithms for this input sequence clearly indicating the values of the parameters for each call. **[3 M]**