III/3rd Sem CS 2009

15/12

* 3 E - 1 4 6 3 / 6 4 4 0 *

Total Printed Pages : 7

3E-1463

Roll No. : _____

1463

B. E. - II Year (Sem. III) Examination, December – 2007

Data Structures & Algorithms

(Common to Computer Engg. & IT)

Time : 3 Hours] [Total Marks : 80

[Min. Passing Marks : 24

*Attempt overall **5** questions. Selecting **one** question from each unit.*
*All questions carry **equal** marks.*

Use of following supporting material is permitted during examination. (Mentioned in form No. 205)

1. Nil 2. Nil

UNIT – I

1 (a) Which function grows faster ?

(i) $n^{\log n}; (\log n)^n$

(ii) $\log n^k; (\log n)^k$

(iii) $n^{\log\log\log n}; (\log n)!$

(iv) $n^n; n!$

4

(b) If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$ where f_1 and f_2 are positive functions of n, show that the function $f_1(n)+f_2(n)$ is $O(\max(g_1(n), g_2(n)))$.

4

(c) A linked list has exactly n nodes. The elements in these nodes are selected from the set $\{0, 1, \ldots, n\}$. There are no duplicates in the list.

Design an $O(n)$ worst case time algorithm to find which one of the elements from the above set is missing in the given linked list. 8

OR

1 (a) Consider an array of size n. Sketch an $O(n)$ algorithm to shift all items in the array k places cyclically counterclockwise. You are allowed to use only one extra location to implement swapping of items. 8

(b) Let A be an array of size n, containing positive or negative integers, with $A[1] < A[2] < \ldots < A[n]$

Design an efficient algorithm (should be more efficient than $O(n)$) to find an i such that $A[i] = i$ provided such as i exists. What is the worst case computational complexity of your algorithm ? 8

UNIT – II

2 (a) A stack can be used to resolve the "scope of an identifier" within a program. Some languages permit using the same identifier name for local and global identifiers. Write an algorithm for

(i) Determining whether an identifier has been used within the scope and

(ii) referring to a local variable if the same identifier has been defined as local as well as global.

Use C/C++ as reference language. 8

(b) A queue Q contains the items $a_1, a_2, \ldots, a_n$ in that order with a_1 at the front and a_n at the back. It is required to transfer these items on to a stack S (initially empty) so that a_1 is at the top of the stack and the order of all other items is preserved. Using insert_queue ( ) and delete_queue ( ) operations for the queue and push ( ) and pop( ) operations for the stack, outline an efficient $O(n)$ algorithm to accomplish the above task, using only a constant amount of additional storage. 8

OR

2 (a) A stack is used to implement recursion within a program. Explain how the calls are tracked and finally programs exit from recursion ? Take the example of Fibonacci Series. 6

(b) A priority queue is implemented as multiple queues with one queue for a range of priorities using the following rule :
Assign a job of priority i to i^{th} queue if the jobs in the queue is less than k else assign it to $(i+1)$ th queue.
Write an algorithm to
(i) Create the priority queue
(ii) Insert an element into the queue. 10

UNIT – III

3 (a) Suppose that we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequence could not be the sequence of nodes examined ? Give reason.
(i) 2, 252, 401, 398, 330, 344, 397, 363
(ii) 924, 220, 911, 244, 898, 258, 362, 363
(iii) 925, 202, 911, 240, 912, 245, 363
(iv) 2, 399, 387, 219, 266, 382, 381, 278, 363
(v) 935, 278, 347, 621, 299, 392, 358, 363 10

(b) Devise an algorithm that takes two values, a and b such $a \leq b$, and visits all keys x in a Binary search tree such that $a \leq x \leq b$. The running time of your algorithm should be $O(N + \log n)$ where N is the number of keys visited and n is the number of keys in the tree. 6

OR

3 (a) In a binary search tree, given a key x, define successor (x) as the key which is the successor of x in the sorted order determined by an inorder tree walk. Define predecessor (x) similarly. Given x, write an algorithm to find its successor and predecessor. 8

(b) A random binary search tree having n nodes, where $n = 2^k - 1$, for some positive integer k, is to be reorganized into a perfectly balanced binary search tree. Outline an efficient algorithm for this task. The algorithm should not use any memory locations other than those already used by the random binary search tree. 8

UNIT – IV

4 (a) An Euler Tour of a connected graph $G = (V, E)$ is a cycle that traverses each edge of G exactly once, although it may visit a vertex more than once. For an Euler cycle to exist, the degree of each node must be even. Write an algorithm to determine whether the given graph is Euler when.

(i) Adjacency Matrix is used to represent the graph.

(ii) Adjacency List is used to represent the graph. 6

(b) An activity graph is a representation of input and output relationship in the form of a directed graph. Suppose such a graph is inputted as below :

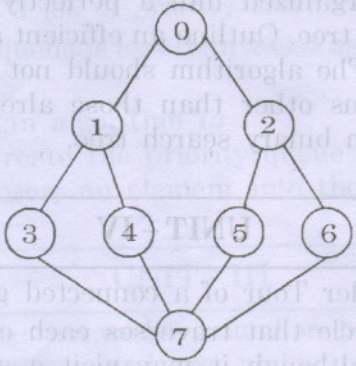
Node	Inputs to the node is provided by
Z	B, C, D
X	A, B, Y

Write an algorithm to create the adjacency list representation of the graph. 10

OR

4 (a) Let T be a tree with root. The edges of T are undirected. Each edge in T has a nonnegative length. Write an algorithm to determine the length of the shortest paths from root to the remaining vertices of T. 8

(b) Find the adjacency matrix and adjacency list representation for the following graph :



Also list the order of traversal when traversed in (i) BFS and (ii) DFS order. 8

UNIT - V

5 (a) Design an efficient algorithm to find all duplicates in a list of n elements. 8

(b) A sorting algorithm is said to be stable if it preserves the original order to records with equal keys. Suppose we have an array of n data records such that the key of each record has the value 0, 1, or 2. Outline a worst case linear time algorithm to sort these records in place, using only an additional amount of storage equal to that of one record. Is your algorithm stable ? Justify. 8

OR

5	(a)	Write an algorithm to find first location of a character in an array of characters represented as an array of array of characters. (Pointer to pointer)	8
	(b)	Suppose we have a sorted array of strings $s_1, \ldots s_n$. Write an algorithm to determine whether a given string x is a member of this sequence. What is the time complexity of your algorithm as a function of n and the length of x ?	8