Exam No: _____

GANPAT UNIVERSITY
M. TECH. SEMESTER – II COMPUTER ENGINEERING
REGULAR EXAMINATION JUN - 2012
3CE202: COMPUTER ALGORITHMS

TIME:-3 HOURS] **[TOTAL MARKS: 70**

Instructions:
1. Figures to the right indicate full marks.
2. Each section should be written in a separate answer book.
3. Be precise and to the point in your answer.

SECTION – I

Q – 1 **(A)** Let t_A (n) and t_B (n) denote the running times of two programs A and B respectively. For following pairs find the value of n for which program A is faster than program B. **[4]**

1. $t_A(n) = 2^n$, $t_B(n) = 100n$
2. $t_A(n) = 1000n$, $t_B(n) = n^2$

(B) Express complexity of following functions using theta (Θ) notation. Clearly indicates value of constants C_1, C_2 and n_0. **[8]**

1. $f(n) = \sqrt{n} + n^2$
2. $f(n) = 3^{(n+2)} + 2n^2$
3. $f(n) = n^{2.5} + 5n(\lg n)$
4. $f(n) = 5n^3 - n^2 + \lg n$

OR

Q – 1 **(A)** Construct the 3-way B-tree on following data: **[4]**
1, 22, 35, 64, 16, 10, 3, 6, 15, 70, 25, 40

(B) Prove followings: **[8]**

1. If $P(n) = a_0 + a_1 n + a_2 n^2 + \ldots\ldots + a_m n^m$ then Prove that $P(n) = \Theta(n^m)$
2. $2^{n+1} = O(2^n)$ but $2^{2n} \neq O(2^n)$

Q– 2 **Solve following recurrence relations and express your answer using big-oh (O) notation.**

(A) **[5]**

$$T(n) = \begin{cases} 0 & \text{, if } n=0 \\ 5 & \text{, if } n=1 \\ 3T_{n-1} + 4T_{n-2} & \text{, if } n>1 \end{cases}$$

(B) $t_n = 2t_{n-1} + (n+5)3^n$ **[6]**

OR

Q – 2 **Solve following recurrence relations and express your answer using big-oh (O) notation.**

(A) $f(n) = n^2 + f(n-1)$ **[3]**

(B) $T(n) = 3T(n/4) + n \lg(n)$ **[4]**

(C) $T(n) = \begin{cases} n & \text{, if } n=0 \text{ or } n=1 \\ T_{n-1} + T_{n-2} & \text{, if } n>1 \end{cases}$ [4]

Q – 3 **(A)** Arrange following growth rates in the increasing order. [2]
$O(n^3)$, $O(n^{0.5})$, $O(n \lg n)$, $O(n^{1.001})$, $\Omega(n^2)$

(B) Write an algorithm to calculate p^n. Algorithm should have complexity less than $O(n)$ complexity. [5]

(C) Draw the DFS and BFS tree for graph given in fig A. [5]

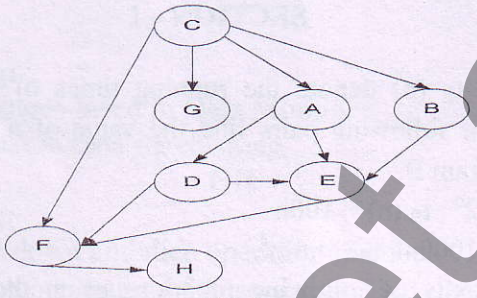

Fig A.

SECTION – II

Q – 4 **(A)** Show the tracing of input instance for which complexity of quick sort is O (n^2). Also write its recurrence relation. Suggest change in quick sort to improve its time complexity. **[6]**

(B) Find minimum spanning tree from graph given in fig B using kruskal's algorithm. **[6]**

OR

Q – 4 **(A)** What is memoization? Explain memoized version of Fibonacci series. **[6]**

(B) Explain backtracking using 4-queen problem. **[6]**

Q – 5 **(A)** Write binary search algorithm and give it's best and worst case time complexity. **[4]**

(B) Write an algorithm for making change problem using dynamic programming. Also construct the dynamic programming table to give change of 9 with coins of denomination 1, 5, and 8. **[7]**

OR

Q – 5 **(A)** Let $A=i_1, i_2, i_3 \ldots i_k$ be the intervals return by minimum ending time first strategy of greedy algorithm and $O=j_1, j_2, j_3 \ldots j_m$ be the intervals return by optimal algorithm then prove that greedy return the optimum answer. **[5]**

(B) Discuss graph representation techniques with example. Also discuss space complexity for each technique. **[6]**

Q – 6 Find the optimal way of multiplying following matrices using dynamic programming. **[12]**

P: 10 x 5, **Q:** 5 x 60, **R:** 60 x 30, **T:** 30 x 5, **U:** 5 x 20.

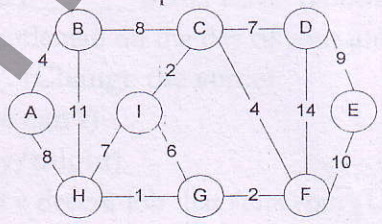

Fig B.

END OF PAPER