

Maulana Azad National Urdu University
B.Tech V Semester Examination - December - 2018
Paper - BTCS502PCT : Design Analysis and Algorithm

پرچہ : ڈیزائن انالیسیز اینڈ الگورتھم

Time : 3 hrs

Marks : 70

ہدایات:

یہ پرچہ سوالات تین حصوں پر مشتمل ہے: حصہ اول، حصہ دوم، حصہ سوم۔ ہر جواب کے لئے لفظوں کی تعداد اشارہ ہے۔ تمام حصوں سے سوالوں کا جواب دینا لازمی ہے۔

1. حصہ اول میں 10 لازمی سوالات ہیں جو کہ معروضی سوالات/خالی جگہ پُر کرنا/مختصر جواب والے سوالات ہیں۔ ہر سوال کا جواب لازمی ہے۔ ہر سوال کے لیے 1 نمبر مختص ہے۔ (10 x 1 = 10 Marks)

2. حصہ دوم میں 8 سوالات ہیں، اس میں سے طالب علم کو کوئی پانچ سوالوں کے جواب دینے ہیں۔ ہر سوال کا جواب تقریباً دو سو (200) لفظوں پر مشتمل ہے۔ ہر سوال کے لیے 6 نمبرات مختصر ہیں۔ (5 x 6 = 30 Marks)

3. حصہ سوم میں 5 سوالات ہیں۔ اس میں سے طالب علم کو کوئی تین سوالوں کے جواب دینے ہیں۔ ہر سوال کا جواب تقریباً پانچ سو (500) لفظوں پر مشتمل ہے۔ ہر سوال کے لیے 10 نمبرات مختص ہیں۔ (3 x 10 = 30 Marks)

حصہ اول

سوال (1)

- (i) ان میں سے سب سے (Slowest) سست ترین Sorting Procedure کون سا ہے؟
 (a) Quick Sort (b) Merge Sort (c) Bubble Sort (d) ان میں سے کوئی نہیں
- (ii) Merge Sort کی Worst Case Time Complexity..... ہے۔
 (a) $O(n^2)$ (b) $O(\log n)$ (c) $O(n)$ (d) $O(n \log n)$
- (iii) 0-1 Knapsack Problem کو Greedy Algorithm کے ذریعہ Solve کیا جاسکتا ہے؟
 (a) True (b) False
- (iv) Floyd Warshall All Pair Shortest Path Algorithm..... میں Pair of Nodes کے بیچ Shortest Path معلوم کرتا ہے۔
 (a) $O(\log n)$ (b) $O(n^2)$ (c) $O(mn)$ (d) $O(n^3)$
- (v) Dijkshtra Algorithm کو..... Shortest Path Problem کہا جاتا ہے۔
 (a) Multiple Source (b) Single Source
 (c) Single Destination (d) Multiple Destination
- (vi) Knapsack Problem..... کی مثال ہے۔
 (a) Greedy Algorithm (b) Dynamic Programming
 (c) Divide & Conquer (d) ان میں سے کوئی نہیں

(vii) ان میں سے کون سا Backtracking الگورتھم نہیں ہے۔

N Queen Problem (b)

Knight Tour Problem (a)

M Coloring Problem (d)

Tower of Hanoi (c)

(viii) اگر $p! = Np$ مان لیا جائے تو ان میں سے کون سا صحیح ہے۔

$Np - \text{hard} = Np$ (b)

Np Complete Np (a)

ان میں سے کوئی نہیں (d)

$p = Np$ Complete (c)

(ix) 8-Queens Problem کو حل کرنے کے لیے کون سا Algorithm استعمال ہوتا ہے۔

Back Tracking (d)

Branch & Band (c)

Dynamic (b)

Greedy (a)

(x) ان میں سے کون سا $O(n^2)$ نہیں ہے۔

$(2^{20}) * n$ (d)

$n^3 / (\text{sqrt}(n))$ (c)

$n^{1.98}$ (b)

$(15^{10}) * n + 12099$ (a)

حصہ دوم

(2) Merge Sort استعمال کرتے ہوئے مندرجہ List کو Sort کیجیے۔ (38, 27, 43, 39, 82, 100)؟

(3) Binary Search Algorithm کی Average Case Complexity معلوم کیجیے۔

(4) دیے گئے Knapsack Instance $n=3$ $m=20$ اور $w_1, w_2, w_3 = 18, 15, 10$ اور $P_1, P_2, P_3 = 25, 24, 15$

اس کا Optimal Solution معلوم کیجیے۔

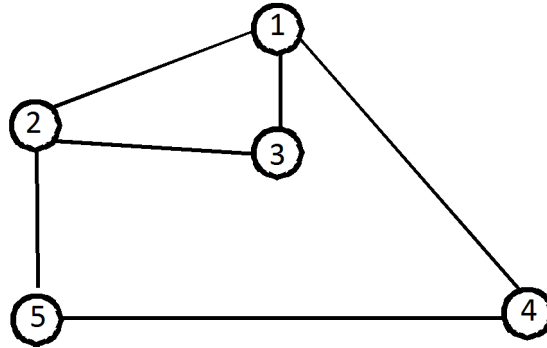
(5) $n=4$ جس کے K_1, \dots, K_4 Keys ہے اور $(I_1, I_2, I_3, I_4) = (5, 6, 7, 8, 20)$ Items ہے اور جس کی

Search Frequency $(q_1, q_2, q_3, q_4) = (4, 2, 6, 3)$ ہے اس کا Optimal Binary Search Tree (OBST) معلوم کیجیے۔

(6) $(w_1, w_2, w_3) = (2, 3, 3)$ اور $(p_1, p_2, p_3) = (1, 2, 4)$ $M=6$ ہے اسے غور کیجیے Dynamic Programming کو

استعمال کرتے ہوئے دیے گئے Knapsack Problem کو حل کیجیے اور Algorithm کو سمجھائیے۔

(7) دیے گئے Graph کو Coloring کرنے کے لیے Back Tracking Method سمجھائیے۔



(8) P اور Np - Class Problem کیا ہے اور اس کو موزوں مثال کے ذریعہ سمجھائیے۔

(9) LIFOBB with 4 Digits کو استعمال کرتے ہوئے دیے گئے Sum of Subset Problems کے Problem Instances کا حل

معلوم کیجیے۔

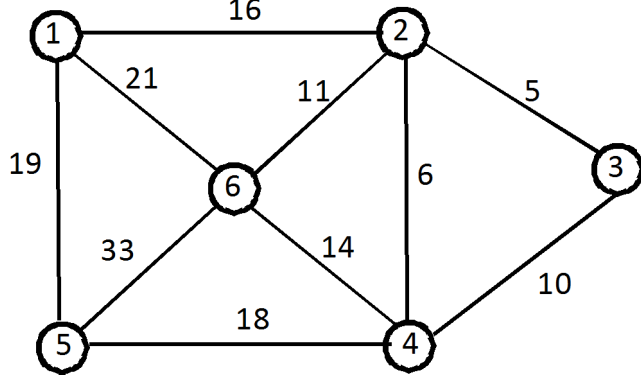
$M = 13$ اور $n = 4$ $(w_1, w_2, w_3, w_4) = (7, 4, 9, 6)$

حصہ سوم

(10) FIFOBB کو استعمال کرتے ہوئے Four Queens Problems کو حل کیجیے۔

(11) Quick Sort کا Best Case اور Worst Case Complexity معلوم کیجیے اور Algorithm کو سمجھائیے۔

(12) Prim's Algorithm کو استعمال کرتے ہوئے دیے گئے Graph کا Minimum Spanning Tree معلوم کیجیے۔

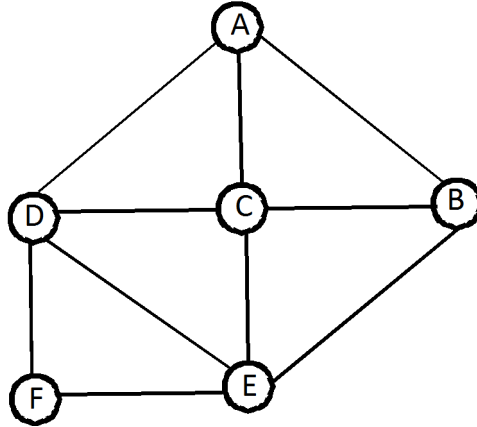


(13) Dynamic Programming کو استعمال کرتے ہوئے دیے گئے Travelling Sales Person Problem کو حل کیجیے۔

$$\begin{bmatrix} \infty & 8 & 6 & 12 \\ 3 & \infty & 9 & 6 \\ 6 & 7 & \infty & 5 \\ 2 & 4 & 11 & \infty \end{bmatrix}$$

(14) Hamiltonian Cycle کے بارے میں بحث کیجیے۔ دیے گئے Graph کو نوٹ کرتے ہوئے بتائیے کہ کیا اس میں Hamiltonian Cycle

Exist کرتی ہے۔



☆☆☆