

**Sample Question Format**

**(For all courses having end semester Full Mark=50)**

**KIIT Deemed to be University**

**Online End Semester Examination(Autumn Semester-2021)**

**Subject Name & Code: Design & Analysis of Algorithms (CS-2012)**

**Applicable to Courses:CSE, IT, CSCE, CSSE, ECS**

**Full Marks=50** **Time:2 Hours**

**SECTION-A(Answer All Questions. Each question carries 2 Marks)**

**Time:30 Minutes (7×2=14 Marks)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question No** | **Question Type(MCQ/SAT)** | **Question** | **CO Mapping** | **Answer Key**  **(For MCQ Questions only)** |
| **Q.No:1** | **MCQ** | Let B(n), W(n) and A(n) denote the best case, worst case and average case running time of an algorithmrespectively, executed on an input of size n. Which of the following is always TRUE?   1. B(n) = O(W(n)) 2. W(n) = Θ(A(n)) 3. A(n) = O(B(n)) 4. A(n) = (W(n)) 5. NONE OF THE OPTION | **A** | **CO1** |
|  | **MCQ** | Let B(n), W(n) and A(n) denote the best case, worst case and average case running time of an algorithm respectively, executed on an input of size n. Which of the following is always TRUE?   1. B(n) = Θ(W(n)) 2. W(n) = (A(n)) 3. A(n) = O(B(n)) 4. B(n) = (W(n)) 5. NONE OF THE OPTION | **B** | **CO1** |
|  | **MCQ** | Let B(n), W(n) and A(n) denote the best case, worst case and average case running time of an algorithm respectively, executed on an input of size n. Which of the following is NOT always TRUE?   1. B(n) = O(W(n)) 2. W(n) = (A(n)) 3. A(n) = (B(n)) 4. B(n) = (W(n)) 5. NONE OF THE OPTION | **D** | **CO1** |
|  | **MCQ** | Let B(n), W(n) and A(n) denote the best case, worst case and average case running time of an algorithm respectively, executed on an input of size n. Which of the following is NOT always TRUE?   1. B(n) = O(W(n)) 2. W(n) = (A(n)) 3. A(n) = Θ(B(n)) 4. W(n) = Θ(W(n)) 5. NONE OF THE OPTION | **C** | **CO1** |
| **Q.No:2** | **MCQ** | Consider the following function.  int fun (int n)  {  if (n == 0 || n==1)  return n;  else  return 1 + fun(n-1) + fun(n-1);  }  What is the least upper bound time complexity of the function fun?   1. O(n) 2. O(nlog n) 3. O(n2) 4. O(2n) 5. NONE OF THE OPTION | **D** | **CO3** |
|  | **MCQ** | void fun(int n, int A[])  {  int i=0, j=0;  while(i < n){  while(j < n &&A[i] <A[j])  { j++;}  i++;  }  }  What is the least upper bound time complexity of the function fun?   1. O(n) 2. O(nlog n) 3. O(n2) 4. O(2n) 5. NONE OF THE OPTION | **A** | **CO3** |
|  | **MCQ** | Consider the following C function  int fun(intn, int A[])  {  int i, j, s=0;  for(i = 1; i<= n; i++)  {  for(j=1; j<n; j += i)  {s = s + A[i];}  }  retun s;  }  What is the least upper bound time complexity of the function fun?   1. O(n) 2. O(nlog n) 3. O(n2) 4. O(2n) 5. NONE OF THE OPTION | **B** | **CO3** |
|  | **MCQ** | Consider the following C function  int fun(intn, int A[])  {  int i, j, s=0;  for(i = 1; i<= n; i++)  {  j = n;  while (j> 0)  {  s= s +j;  j = j - 2;  }  }  return s;  }  What is the least upper bound time complexity of the function fun?   1. O(n) 2. O(nlog n) 3. O(n2) 4. O(2n) 5. NONE OF THE OPTION | **C** | **CO3** |
| **Q.No:3** | **MCQ** | Let Array A[1..11]={9, 5, 7, 3, 2, 6, 7, 3, 1, 2, 1} is a max-heap. What will be resultant max-heap, if the value at index 5 is changed to 7.   1. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 1} 2. {9, 5, 7, 3, 3, 6, 7, 3, 1, 2, 1} 3. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 2} 4. {9, 7, 7, 5, 2, 6, 7, 3, 1, 2, 1} 5. NONE OF THE OPTION | **A** | **CO4** |
|  | **MCQ** | Let Array A[1..11]={9, 5, 7, 3, 2, 6, 7, 3, 1, 2, 1} is a max-heap. What will be resultant max-heap, if the value at index 5 is changed to 3.   1. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 1} 2. {9, 5, 7, 3, 3, 6, 7, 3, 1, 2, 1} 3. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 2} 4. {9, 7, 7, 5, 2, 6, 7, 3, 1, 2, 1} 5. NONE OF THE OPTION | **B** | **CO4** |
|  | **MCQ** | Let Array A[1..11]={9, 5, 7, 3, 2, 6, 7, 3, 1, 2, 1} is a max-heap. What will be resultant max-heap, if the value at index 11 is changed to 7.   1. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 1} 2. {9, 5, 7, 3, 3, 6, 7, 3, 1, 2, 1} 3. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 2} 4. {9, 7, 7, 5, 2, 6, 7, 3, 1, 2, 1} 5. NONE OF THE OPTION | **C** | **CO4** |
|  | **MCQ** | Let Array A[1..11]={9, 5, 7, 3, 2, 6, 7, 3, 1, 2, 1} is a max-heap. What will be resultant max-heap, if the value at index 8 is changed to 7.   1. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 1} 2. {9, 5, 7, 3, 3, 6, 7, 3, 1, 2, 1} 3. {9, 7, 7, 3, 5, 6, 7, 3, 1, 2, 2} 4. {9, 7, 7, 5, 2, 6, 7, 3, 1, 2, 1} 5. NONE OF THE OPTION | **D** | **CO4** |
| **Q.No:4** | **MCQ** | Consider the following graph.    Among the following sequences, which are possible breadth first traversals of the above graph if the first symbol of each sequence is considered as start vertex.  i) BMATHEC ii) MTECHBA iii) ABTHCME iv)ECTMHBA   1. i, ii & iii 2. ii & iii 3. i, ii & iv 4. i & iii 5. NONE OF THE OPTION | **C** | **CO4** |
|  | **MCQ** | Consider the following graph.    Among the following sequences, which are possible breadth first traversals of the above CO4graph if the first symbol of each sequence is considered as start vertex.  i) MBCTHAE ii) MTCHBEA iii) ABTHCME iv) ETCMHBA   1. i &ii 2. ii & iii 3. iii & iv 4. i & iv 5. NONE OF THE OPTION | **A** | **CO4** |
|  | **MCQ** | Consider the following graph.    Among the following sequences, which are possible breadth first traversals of the above graph if the first symbol of each sequence is considered as start vertex.  i) MBCTHEA ii) MTCHBEA iii) ABTHCME iv) ETCMBAH   1. i&ii 2. ii & iii 3. iii & iv 4. ii & iv 5. NONE OF THE OPTION | **D** | **CO4** |
|  | **MCQ** | Consider the following graph.    Among the following sequences, which are possible breadth first traversals of the above graph if the first symbol of each sequence is considered as start vertex.  i) MBCTHEA ii) MTCHBEA iii) HCMAETB iv)HCETMBA   1. i &ii 2. iii & iv 3. i & iv 4. ii & iii 5. NONE OF THE OPTION | **D** | **CO4** |
| **Q.No:5** | **MCQ** | Let A1, A2, A3 and A4 be four matrices of dimensions 3x2, 2x4, 4x5, 5x6 respectively. The number of scalar multiplications required to find the product like ((A1A2)A3)A4) is \_\_\_\_\_\_\_ using the basic matrix multiplication method.   1. 165 2. 170 3. 174 4. 180 5. NONE OF THE OPTION | **C** | **CO2** |
|  | **MCQ** | Let A1, A2, A3 and A4 be four matrices of dimensions 6x2, 2x4, 4x5, 5x3 respectively. The number of scalar multiplications required to find the product like ((A1A2)(A3A4)) is \_\_\_\_\_\_\_ using the basic matrix multiplication method.   1. 165 2. 170 3. 174 4. 180 5. NONE OF THE OPTION | **D** | **CO2** |
|  | **MCQ** | Let A1, A2, A3 and A4 be four matrices of dimensions 3x2, 2x4, 4x5, 5x5 respectively. The number of scalar multiplications required to find the product like (A1(A2(A3A4))) is \_\_\_\_\_\_\_ using the basic matrix multiplication method.   1. 165 2. 170 3. 174 4. 180 5. NONE OF THE OPTION | **B** | **CO2** |
|  | **MCQ** | Let A1, A2, A3 and A4 be four matrices of dimensions 5x2, 2x4, 4x5, 5x3 respectively. The number of scalar multiplications required to find the product like ((A1(A2A3))A4) is \_\_\_\_\_\_\_ using the basic matrix multiplication method.   1. 165 2. 170 3. 174 4. 180 5. NONE OF THE OPTION | **A** | **CO2** |
| **Q.No:6** | **MCQ** | Let the problem X ϵ NP. Which of the following statements are TRUE?   1. If X satisfies NP-hard condition, problem X is NP-complete. 2. There is a polynomial time algorithm for X. 3. X can be verified in polynomial time. 4. i, ii and iii 5. i, iii 6. ii, iii 7. i, ii 8. None of the options | **B** | **CO5** |
|  | **MCQ** | Let the problem A ϵ NP and B ϵ NP Hard. Which of the following statements are TRUE?   1. There is a polynomial time algorithm for A. 2. There is no polynomial time algorithm for B. 3. Problem A may belongs to NPC 4. i, ii and iii 5. i, iii 6. ii, iii 7. i, ii 8. None of the options | **C** | **CO5** |
|  | **MCQ** | Let the problem X ϵ NP. Which of the following statements are TRUE?   1. If X satisfies NP-hard condition, problem X is NP-complete. 2. There is no polynomial time algorithm for X. 3. X cannot be verified in polynomial time. 4. i, ii and iii 5. i, iii 6. ii, iii 7. i, ii 8. None of the options | **D** | **CO5** |
|  | **MCQ** | Let the problem X ϵ NP and Y ϵ NPC. Which of the following statements are TRUE?   1. Y ϵ NP 2. There is no polynomial time algorithm for X. 3. X can be verified in polynomial time. 4. i, ii and iii 5. i, iii 6. ii, iii 7. i, ii 8. None of the options | **A** | **CO5** |
| **Q.No:7** |  | Match the following pairs:   1. O(log n) i. Worst case Quick Sort 2. O(n) ii. Binary Search 3. O(n log n) iii.Best Case Insertion Sort 4. O(n2) iv. Merge Sort   v.Linear Search   1. P-ii, Q-iii, Q-v, R-iv, S-i, 2. P-iii, Q-ii, R-iv, R-i, S-i 3. P-i, Q-ii, R-iv, S-iii, S-i 4. P-iv, P-v, Q-ii, R-i, S-iii 5. NONE OF THE OPTION | **A** | **CO3** |
|  | **MCQ** | Match the following pairs:   1. O(1) i. Best case Insertion Sort 2. O(n) ii. Best case Linear Search 3. O(n log n) iii.Worst case Bubble Sort 4. O(n2) iv. Heap Sort   v.Merge Sort   1. P-ii, Q-iii, R-iv, S-i, S-iii 2. P-iii, Q-ii, Q-v, R-iv, S-i 3. P-ii, Q-i, R-iv, R-v, S-iii 4. P-iv, Q-ii, Q-v, R-i, S-iii 5. NONE OF THE OPTION | **C** | **CO3** |
|  | **MCQ** | Match the following pairs:   1. Floyd-Warshall i. Divide-and-Conquer   Algorithm   1. Quick Sort ii. Greedy Approach 2. Fractional iii.Dynamic Programming   Knapsack  problem   1. O(n) iv. Linear Search   v.Best Case Insertion Sort   1. P-ii, Q-iii, R-iv, R-v, S-i 2. P-iii, Q-i, R-ii, S-iv, S-v 3. P-ii, P-v, Q-i, R-iv, S-iii 4. P-iv, Q-ii, R-i, S-iii, S-v 5. NONE OF THE OPTION | **B** | **CO3** |
|  | **MCQ** | Match the following pairs:   1. Dijkestrals i. Divide-and-Conquer   Algorithm   1. LCS ii. Greedy Approach 2. Merge Sort iii.Dynamic Programming 3. O(nlog n) iv.Merge Sort   v.Heap Sort   1. P-ii, Q-iii, R-iv, S-iv, S-v 2. P-iii, Q-i, R-ii, R-v, S-iv 3. P-ii, Q-i, R-iv, S-iv, S-v 4. P-ii, Q-iii, R-i, S-iv, S-v 5. NONE OF THE OPTION | **D** | **CO3** |

**SECTION-B(Answer Any Three Questions. Each Question carries 12 Marks)**

**Time: 1 Hour and 30 Minutes** **(3×12=36 Marks)**

|  |  |  |
| --- | --- | --- |
| **Question No** | **Question** | **CO Mapping**  **(Each question should be from the same CO(s))** |
| **Q.No:8** | Consider the following function  int fun( int n)  {  if (n<=1)  return 1;  else  return n + fun(n-1) + fun(n-2);  }   1. What does the above function compute? 2. Executing the function for n=5+d%4 (where d is the last digit of your roll number). Draw a recurrence tree to illustrate this fact. 3. How many additions are performed to compute fun(n), where n=5+d%4 and d represents the last digit of your roll number? 4. Assuming that each addition/subtraction takes constant time, write a recurrence relation for the running time of fun(n) & solve the recurrence. | **CO1** |
| Consider the following function  int fun( int n)  {  if (n<=1)  return 1;  else  return fun(n-1) + fun(n-2) - n;  }   1. What does the above function compute? 2. Executing the function for n=5+d%3 (where d is the last digit of your roll number). Draw a recurrence tree to illustrate this fact. 3. How many additions& subtractions are performed to compute fun(n), where n=5+d%4 and d represents the last digit of your roll number? 4. Assuming that each addition/subtraction takes constant time, write a recurrence relation for the running time of fun(n) & solve the recurrence. |
| Consider the following function  int fun( int n)  {  if (n<=1)  return 1;  else  return fun(n-1) + 2 + fun(n-2);  }   1. What does the above function compute? 2. Executing the function for n=5+d%4 (where d is the last digit of your roll number). Draw a recurrence tree to illustrate this fact. 3. How many additions& subtractions are performed to compute fun(n), where n=5+d%5 and d represents the last digit of your roll number? 4. Assuming that each addition/subtraction takes constant time, write a recurrence relation for the running time of fun(n) & solve the recurrence. |
| **Q.No:9** | There is a set of n activities with their start and finish times. Assume that the activities are arranged in non-decreasing order of their finish time. Write an algorithm for Activity Selection. The algorithm must give priority in choosing longest duration activity in case of more than one activity having same finish time. | **CO5** |
| There is a set of n activities with their start and finish times. Assume that the activities are arranged in non-decreasing order of their finish time. Write an algorithm for Activity Selection. The algorithm must give priority in choosing shortest duration activity in case of more than one activity having same finish time. |
| There is a set of n activities with their start and finish times. Assume that the activities are arranged in non-decreasing order of their finish time. Write an algorithm for Activity Selection. The algorithm must give priority in choosing the late start activity in case of more than one activity having same finish time. |
| **Q.No:10** | Suppose a file to be transferred through the network contains the following characters with their number of occurrences as < a: 10, b: 25, c: 5, d: 30, e: 20 >. Determine an efficient strategy that can minimize the total cost of transferring that file of 1000 characters. Find out the total cost of transfer if transferring cost for 1-bit of data is 4 units. | **CO2** |
| State and explain the Longest Common Subsequence problem. Determine an LCS of the given two sequences < a, a, b, a, b, a, b, a > and <b, a, b, a, b, b, a, a, b>. |
| Construct the adjacency list of the following directed graph and demonstrate the DFS (depth-first search) algorithm on it. Write the initialization and explain how the relevant parameters and data structures are updated during the execution. In the final step, you should write the DFS tree/trees, and also the forward edges, cross edges, and back edges, if any. Use node 'v' as source node while answering the question.  10-C.png |
| **Q.No:11** | Apply Dijkstra’s algorithm to find the shortest path (values) from vertex ‘a’ to rests of the vertices. Write down the contents of the data structure used to solve the above problem after finding shortest distance to a new vertex. Show the shortest path to all destinations.  (Hints M = Your Roll Number % 5 & N = M \* 2)  11-A.png | **CO3** |
| Apply Dijkstra’s algorithm to find the shortest path (values) from vertex ‘b’ to rests of the vertices. Write down the contents of the data structure used to solve the above problem after finding shortest distance to a new vertex. Show the shortest path to all destinations.  (Hints M = Your Roll Number % 5 & N = M \* 2)  11-B.png |
| Apply Dijkstra’s algorithm to find the shortest path (values) from vertex ‘c’ to rests of the vertices. Write down the contents of the data structure used to solve the above problem after finding shortest distance to a new vertex. Show the shortest path to all destinations.  (Hints M = Your Roll Number % 5 & N = M \* 2)  11-C.png |