**L-2/T-2/CSE** **Date: 11/03/2018**

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-2 B. Sc. Engineering Examinations 2016-2017

Sub: **CSE 207** (Data Structures and Algorithms II)

Full Marks: 210 Time: 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

---

## SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

1. (a) Write the Prim's algorithm that computes a minimum spanning tree of a graph. Analyze the time-complexity of the algorithm. **(15)**

(b) Write the correctness proof of the Kruskal's algorithm that finds a minimum spanning tree of a graph. **(10)**

(c) Draw a minimum spanning tree of the graph $G$ shown in Figure 1 by running (i) Prim's algorithm, and (ii) Kruskal's algorithm. Write the order of the edges in each case (whenever there is a choice of vertices, always use alphabetic ordering). **(10)**

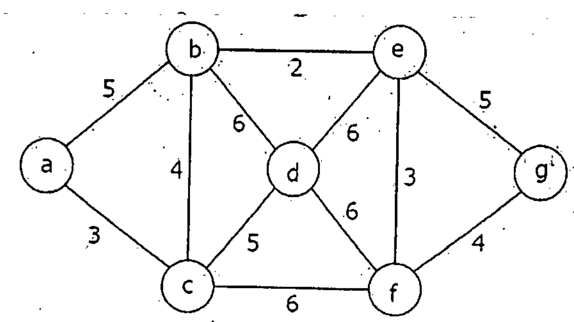



Figure 1: A graph $G$

2. (a) (i) Write the properties that are satisfied by shortest paths of a graph. **(15)**

(ii) Show by an example that the Dijkstra's algorithm gives wrong result for a graph with negative weight.

(iii) Write an algorithm that finds shortest paths in a DAG. Analyze the time-complexity of the algorithm.

(b) Prove that the Bellman-Ford algorithm returns TRUE if the input graph contains no negative-weight cycles reachable from the source, and FALSE otherwise. **(10)**

(c) Compare the key properties of problems that can be solved with greedy method, divide-and-conquer method and dynamic programming method. **(10)**

3. (a) Write the Edmonds-Karp algorithm that solves the maximum-flow problem. Analyze the time-complexity of the algorithm for integral capacities. **(10)**

(b) What is a flow network? Explain how network flow algorithms allow us to find the maximum bipartite matching. **(10)**

Contd .......... P/2

**CSE 207**

**Contd... Q. No. 3**

(c) For the flow network shown in Figure 2, find the value of the maximum flow by drawing residual networks in each step. Finally, draw the maximum flow network. Also draw the residual network and identify the min-cut of the maximum flow network. **(15)**

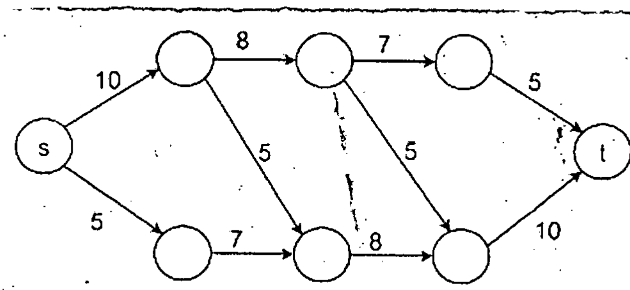



Figure 2: A flow network

4. (a) What is an AVL tree? Draw the AVL tree that results after successively inserting the keys 23, 95, 14, 28, 39, 45, 53, 68, 81, 88, 62 into an initially empty AVL tree. **(10)**

(b) Write the properties of a red-black tree. Show that the height of a red-black tree $T$ storing $n$ items is $O(\log n)$. **(10)**

(c) By writing a procedure, explain the cases that may arise for deleting a node from a red-black tree. **(15)**

## SECTION – B

There are **FOUR** questions in this section. Answer any **THREE**.

5. (a) Assuming simple uniform hashing, prove that a successful search in separate chaining takes expected $O(1+\alpha)$ work, where $\alpha$ is the load factor. **(16)**

(b) Consider inserting the keys 14, 3, 29, 25, 19, 16, 35 and 2 into a hash table of size m = 11 with hash function $h(k) = k \bmod m$. Illustrate the result of inserting these keys using quadratic probing. You don't need to show the intermediate steps, just show the hash table after inserting all the elements. **(8)**

(c) Prove that the first $\lceil m/2 \rceil$ probes in quadratic probing are distinct given m is a prime number. Here m denotes the size of your hash table. **(8)**

(d) What is the major advantage of double hashing over quadratic probing? **(3)**

Contd .......... P/3

6. (a) Suppose $X \in$NP-complete. Prove that $X \in$P if and only if P = NP. **(10)**

(b) Given a set of clauses $C_1, C_2, \ldots, C_k$, each of length 3, over a set of Boolean variables, the 3-SAT problem asks if there exist a satisfying truth assignment. **(18)**

Given a graph G and a number K, the vertex cover problem asks if G contains a vertex cover of size at most K. (For a graph G = (V, E), a set of vertices $S \subseteq V$ is a vertex cover if every edge $e \in E$ has at least one end in S.)

Prove that Vertex Cover problem is at least as hard as 3-SAT problem.

(c) Prove that $\text{NP} \neq \text{co}-\text{NP} \Rightarrow P \neq \text{NP}$. **(7)**

7. (a) Consider the following Boolean formula:

$(x' \vee y') \wedge (x \vee y') \wedge (x' \vee y) \wedge (x \vee y \vee z') \wedge (x \vee y \vee z)$. Use backtracing algorithm to decide whether this is satisfiable or not. You must show the backtracing tree. **(15)**

(b) Consider the following instance of the 0-1 knapsack algorithm. Solve this instance using branch-and-bound algorithm. You must show the branch-and-bound tree. **(20)**

| Item | Weight | Value |
|---|---|---|
| 1 | 2 | 10 |
| 2 | 3 | 9 |
| 3 | 2 | 20 |
| 4 | 5 | 30 |

Krapsack capacity = 10

8. (a) Analyze the running time of the randomized quick sort algorithm. **(13)**

(b) Suppose you have designed a Monte Carlo algorithm for a given problem which succeeds with probability $\geq \frac{1}{n^2}$. However, your boss has asked you to achieve at least $\left(1 - \frac{1}{n}\right)$ success probability. Explain how you can achieve this. **(10)**

(c) Give an approximation algorithm for the vertex cover problem. Analyze the approximation ratio of your algorithm. **(12)**

---------------------------------

**L-2/T-2/CSE** **Date : 30/01/2017**

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-2 B. Sc. Engineering Examinations 2015-2016

Sub : **CSE 207** (Algorithms)

Full Marks: 210 Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

---

## SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

1. (a) What is the matrix chain multiplication problem? Prove that the matrix chain multiplication problem has both the overlapping subproblems property and the optimal substructure property. **(10)**

   (b) (i) Write the steps that are to be followed for solving a problem using dynamic programming method. **(4+6=10)**

   (ii) Let $X = \langle x_1, x_2, \cdots, x_m \rangle$ and $Y = \langle y_1, y_2, \cdots, y_n \rangle$ be sequences, and let $Z = \langle z_1, z_2, \cdots, z_k \rangle$ be any LCS of $X$ and $Y$. If $x_m \neq y_n$ and $z_k \neq x_m$, then prove that $Z$ is an LCS of $X_{m-1}$ and $Y$.

   (c) Find all longest common subsequences of $X = \langle \text{AGGCTGAT} \rangle$ and $Y = \langle \text{GGGCATA} \rangle$ using dynamic programming method. **(15)**

2. (a) (i) Explain why one can solve the fractional knapsack problem by a greedy strategy, but one cannot solve the 0-1 knapsack problem by such a strategy. **(5+5=10)**

   (ii) Find an optimal solution to the fractional knapsack instance of $n = 6$, $M = 40$, $(p_1, p_2, \cdots p_6) = (18, 5, 15, 30, 18, 35)$, and $(w_1, w_2, \cdots w_6) = (9, 5, 10, 10, 12, 10)$.

   (b) (i) Show that the 0-1 knapsack problem has the overlapping subproblems property. **(4+6=10)**

   (ii) Write an algorithm for solving the 0-1 knapsack problem. Analyze the time-complexity of the algorithm.

   (c) (i) Compare backtracking, and branch and bound techniques. **(3+12=15)**

   (ii) Solve the following instance of the 0/1 knapsack problem using the branch-and-bound approach with a state-space-tree. Assume that the knapsack capacity is 15.

| Item | Weight | Value |
|---|---|---|
| 1 | 7 | 140 |
| 2 | 6 | 132 |
| 3 | 4 | 140 |
| 4 | 3 | 90 |

Contd .......... P/2

**CSE 207**

3. (a) Write the Edmonds-Karp algorithm that solves the maximum-flow problem. Analyze the time-complexity of the algorithm for integral capacities. **(10)**

(b) Explain how network flow algorithms allow us to find the maximum bipartite matching. **(10)**

(c) Write an application of prefix sums. What are prefix sums of an array A[$i$], $1 \le i \le n$? Write a parallel algorithm that finds prefix sums of the array A. **(15)**

Illustrate the prefix sums algorithm for the array [15, 35, 40, 25, 20, 50, 10, 30] of eight elements by showing the values of arrays used in bottom-up and top-down traversals.

4. (a) (i) Compare the techniques of brute-force algorithms, heuristics, and approximation algorithms for coping with hard problems. **(5+5=10)**

(ii) Write a backtracking algorithm to color a map with no more than four colors.

(b) Explain the term 'reducibility'. If $L_1$ and $L_2$ are two languages such that $L_1 \le_p L_2$, then show that $L_2 \in$ P. If any NP-complete problem is polynomial-time solvable, then prove that P = NP. **(10)**

(c) (i) What is the vertex-cover problem? Write a polynomial-time approximation algorithm for the vertex-cover problem. Prove the time-complexity and find the approximation ratio of the algorithm. **(7+8=15)**

(ii) For the general traveling-salesman problem, prove that one cannot find good approximate tours in polynomial time unless P = NP.

## SECTION-B

There are **NINE** questions in this section. Answer any **SEVEN**.

5. (a) Define big O-notation, $\Omega$-notation and $\Theta$-notation. **(6+9=15)**

(b) For each of the following pairs indicate whether $f(n)$ is O(g($n$)), $\Omega$(g($n$)) or both (i.e. $\Theta$(g($n$))). Provide brief justifications.

| $f(n)$ | $g(n)$ |
|---|---|
| $\sqrt{n}$ | $\log_2 n$ |
| $n^{1.01}$ | $n \log_2^2 n$ |
| $2^n$ | $2^{n+1}$ |
| $n!$ | $2^n$ |
| $n^{100}$ | $1.01^n$ |
| $\sum_{i=1}^{n} i^k$ | $n^{k+1}$ |

## CSE 207

6. Suppose you are choosing among the following three (possibly hypothetical) algorithms for multiplying two $n \times n$ matrices: **(15)**

   - Algorithm A solves the problem by dividing each $n \times n$ matrix into four $n/2 \times n/2$ matrices, make 8 recursive calls to compute intermediate results and then combine them in $\Theta(n^2)$ time to get the result.
   - Algorithm B (Strassen's algorithm) solves the problem by dividing each $n \times n$ matrix into four $n/2 \times n/2$ matrices, make 7 recursive calls to compute intermediate results and then combine them in $\Theta(n^2)$ time to get the result.
   - Algorithm C solves the problem by dividing each $n \times n$ matrix into nine $n/3 \times n/3$ matrices, make 9 recursive calls to compute intermediate results and then combine them in $\Theta(n^2)$ time to get the result.

   Use the master theorem to determine asymptotic running times of the three algorithms. Which one would you choose?

7. Counting Inversions: This problem arises in the analysis of rankings. Consider comparing two rankings. One way is to label the elements (books, movies, etc.) from 1 to $n$ according to one of the rankings then order these labels according to the other ranking, and see how many pairs are "out of order". **(15)**

   We are given a sequence of n distinct numbers $a_1, \cdots, a_n$. We say that two indices $i < j$ form an inversion if $a_i > a_j$ that is if the two elements $a_i$ and $a_j$ are "out of order". Provide a divide and conquer algorithm to determine the number of inversions in the sequence $a_1, \cdots, a_n$ in time O($n \log n$).

   (Hint: Modify merge sort to count during merging)

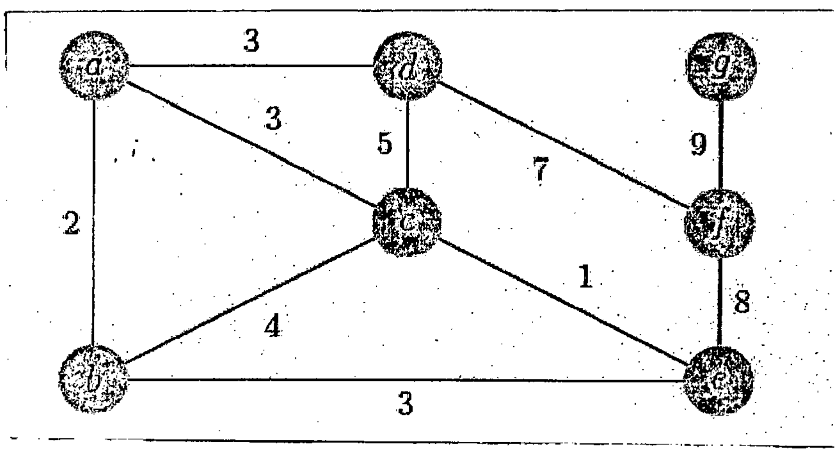
8. Given an undirected graph, $G = (V, E)$, provide a linear time algorithm to check whether there is a cycle of odd length in the graph. Briefly justify its correctness. **(15)**

9. Suppose a CS curriculum consists of $n$ courses, all of them mandatory. The prerequisite directed graph $G$ has a node for each course, and an edge from course $v$ to course $w$ if and only if $v$ is a prerequisite for $w$. Find an algorithm that works directly with this graph representation, and computes the minimum number of semesters necessary to complete the curriculum (assume that a student can take any number of courses in one semester). The running time of your algorithm should be linear. **(15)**

Contd .......... P/4

10. There is a network of roads $G = (V, E)$ connecting a set of cities $V$. Each road in $E$ has an associated length $l_e$. There is a proposal to add one new road to this network, and there is a list $E'$ of pairs of cities between which the new road can be built. Each such potential road $e' \in E'$ has an associated length. As a designer for the public works department you are asked to determine the road $e' \in E'$ whose addition to existing network $G$ would result in the maximum decrease in the driving distance between two fixed cities $s$ and $t$ in the network. Give an algorithm for solving this problem. Your algorithm should run in $O((V + E + E') \log V)$ time. **(15)**

11. Find a minimum spanning tree (MST) in the following graph by running (i) Kruskal' algorithm, and (ii) Prim's algorithm. Show the order of the edges that are selected in each case (whenever there are choices of vertices, always use alphabetic ordering). **(15)**




12. Recall the Interval Scheduling Problem from class. You are given $n$ intervals with the starting and finishing times of $i$-th interval are given by $s_i$ and $f_i$ respectively. A subset of the intervals is compatible if no two of them overlap in time. The goal is to find a compatible set of maximum size. Prove that the greedy strategy of picking an interval, which is compatible with intervals already picked, with smallest finishing time gives an optimal solution. **(15)**

13. (a) Under a Huffman encoding of $n$ symbols with frequencies $f_1, f_2, \cdots, f_n$, what is the longest codeword could possibly be? Give an example set of frequencies that would produce this case. **(7+8=15)**

(b) Following a set of characters and corresponding frequencies:

| Character | Frequency |
|---|---|
| A | 45 |
| B | 13 |
| C | 12 |
| D | 16 |
| E | 5 |
| F | 9 |

Construct Huffman codes for the characters using the greedy algorithm.

---

L-2/T-2/CSE

Date : 26/07/2016

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-2 B. Sc. Engineering Examinations 2014-2015

Sub : **CSE 207** (Algorithms)

Full Marks : 210 Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

---

## SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

1. (a) State the parenthesis theorem for the depth-first search. Prove the theorem. **(10)**

   (b) Find the different types of edges if DFS is applied to the following graph from vertex A. Discuss the edges' properties with respect to the following graph. **(7)**

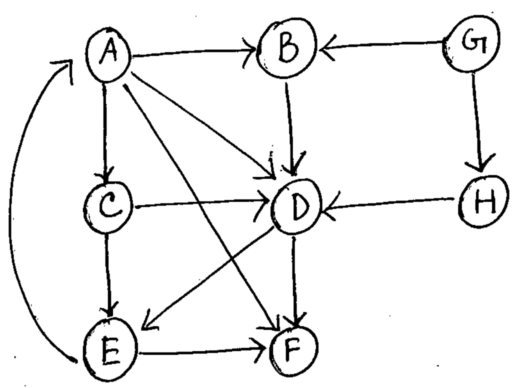



Fig: Q.1 (b)

   (c) Show the topological sorting order of the graph in Question 1(b) with explanation of discovery and finishing time of the node exploration. [Note: There will be no edge from E to A). **(8)**

   (d) Let C and C′ be distinct strongly connected components in directed graph G = (V, E). Suppose that there is an edge (u, v) ∈ E, where u ∈ C and v ∈ C′, then prove that f(C) > f(C′), where f(U) = the latest finishing time of any vertex in U. **(10)**

2. (a) Let G = (V, E) be a connected, undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let (S, V-S) be any cut that 'respects' A and let (u, v) be a "light" edge crossing (S, V-S). Then prove that edge (u, v) is "safe" for A. **(8)**

   (b)

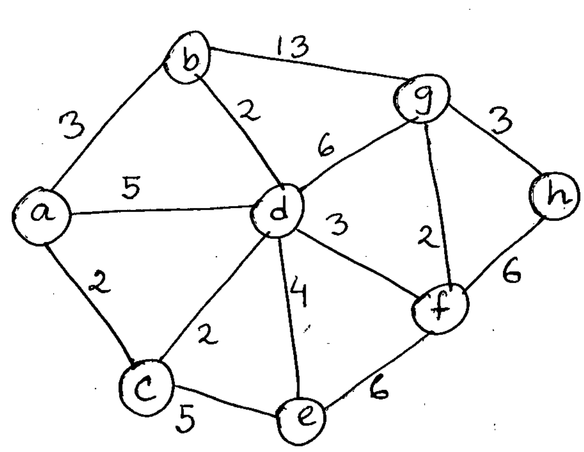



Fig: Q. 2(b)

Contd .......... P/2

## CSE 207

**Contd ... Q. No. 2(b)**

Find the "light" and "respectful" crossing edges for each step of Kruskal's method for building MST of the graph shown in Figure for Q. 2(b). **(12)**

(c) What will be the cost of finding and joining (union) of sets in Kruskal's method? **(5)**

(d) For the graph in Q. 2(b), show the value of u, v.key, Q for the following algorithm (show each step) **(10)**

```
MST (G, w, r)
    Q = V[G]
    for each u ∈ Q
        u.key = ∞
        r.key = 0, p[r] = NULL
        while (Q ≠ empty)
            u = extract min (Q)
            for each v ∈ Adj [u]
                if ((v ∈ Q) and w(u,v) < v.key)
                    p[v] = u
                    v.key = w(u, v)
```

3. (a) State and prove the convergence property of edge relaxation in the context of shortest paths. **(5)**

(b) Provide the correctness proof of the Dijkstra's algorithm. **(10)**

(c) Find the transitive closure for the following graph. Show the matrices computed at each step. What will be the cost to find the transitive clousure? **(10)**

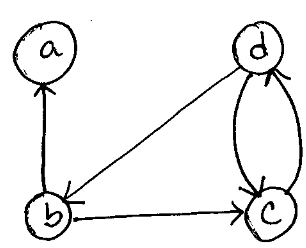



Fig: Q. 3(c)

(d) Apply Belman-Ford algorithm to the following graph to find the shortest paths. Show the relaxation. **(10)**

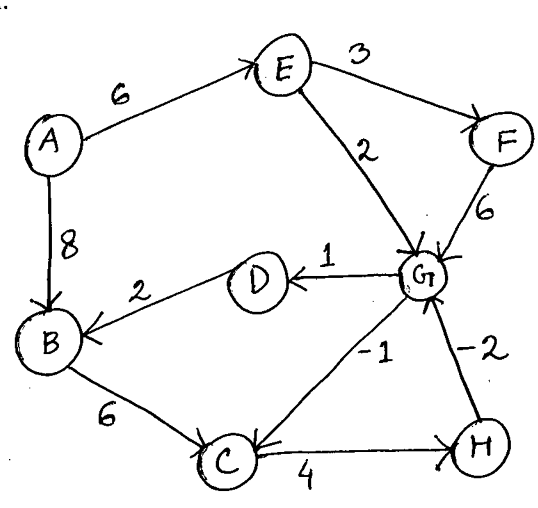



Fig: Q. 3(d)

Contd .......... P/3

**CSE 207**

4. (a) A traffic engineer needs to decide to widen the road for allowing maximum traffic from A to G for the following road-network. Which roads are to be widened? **(12)**

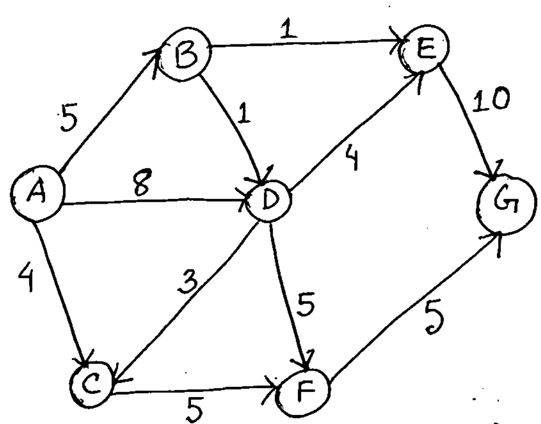



Fig. Q 4 (a)

(b) Define and explain super sink node, augmented path, residual capacity, super source node in a network flow model. **(8)**

(c) What is the pitfall of Ford-Fulkersan's max-flow-min-out method? How can it be improved? **(5)**

(d) Find the maximum job assignment for the following graph. **(10)**

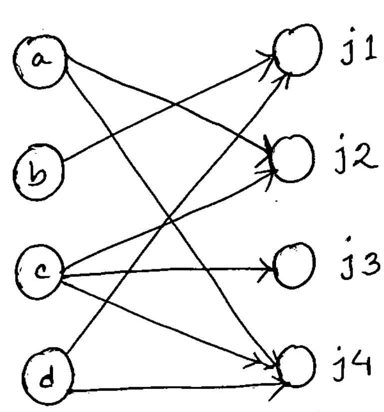



Fig. Q. 4 (d)

**SECTION – B**

There are **FOUR** questions in this section. Answer any **THREE**.

All the symbols have their usual meanings unless explicitly mentioned.

5. (a) What is asymptotic analysis of algorithms? Define big $O$-notation, $\Omega$-notation and $\Theta$-notation. Accordingly, for the following functions: $f(n) = nlogn$ and $g(n) = 10nlog10n$, show whether $f$ is $O(g)$, or $f$ is $\Omega(g)$, or $f$ is $\Theta(g)$. **(3+6+8)**

(b) State master theorem. Explain why master method cannot be used to solve the following recurrence: $T(n) = 2T(n/2) + nlogn$. Using substitution method, prove that $T(n)$ is $O(nlog^2 n)$. **(6+6+6)**

## CSE 207

6. (a) *GreedyCell*, a mobile phone operator, wants to place cell towers (base stations) along a long straight country road, to provide network coverage to $n$ houses sparsely scattered along the road. The distance of these houses from the start of the road are, in miles and in increasing order, $d_1, d_2, \ldots d_n$. Each cell tower has coverage of $R$ miles, i.e., if a house is within $R$ miles of a tower, it would get service from that tower. Now give an efficient algorithm for finding a placement of cell towers that uses as few cell towers as possible to provide coverage to all the $n$ houses. **(12)**

   (b) Given a sorted array of distinct integers $A[1 \ldots n]$, we want to find out whether there is an index $i$ for which $A[i] = i$. Give a divide-and-conquer algorithm that runs in time $O(\log n)$. Write the pseudocode of the algorithm and then prove the runtime based on that code. **(12)**

   (c) Prove that the lower bound on any comparison-based sorting is *nlogn*, i.e., any comparison sort algorithm requires $\Omega(nlongn)$ comparison in the worst case. **(11)**

7. (a) What are the key properties of an optimization problem such that the problem can be solved using dynamic programming? Compare top-down memorized dynamic programming with bottom-up dynamic programming and identify the types of problems for which one outperforms the other. **(4+4)**

   (b) The balanced partition problem is defined as follows. Given a set of $n$ integers, each in the range $[0, N]$, we want to divide the set into two subsets such that difference of sum of the two subsets is minimum. For example, for the set of integers: $S = \{1, 7, 3, 5, 9\}$, a balanced partition will be the subsets, $S1 = \{1, 7, 5\}$ and $S2 = \{3, 9\}$ since for this partition, the difference of sum of numbers on $S1$ and $S2$ is 1 which is the minimum. Now, formulate a dynamic programming solution for the balanced partition problem. Determine the time complexity of your algorithm. **(15)**

   (c) Consider the following instance of the maximum satisfiability (MAX-SAT) problem. **(12)**

   Set of Boolean variables, V = {w, x, y, z}

   Set of clauses, C = {C1, C2, C3, C4, C5, C6}

   where $C1 = (w \vee \bar{x} \vee \bar{y} \vee z)$, $C2 = (w \vee \bar{y})$, $C3 = (x \vee \bar{y})$, $C4 = (y \vee \bar{z})$

   $C5 = (z \vee \bar{w})$, $C6 = (\bar{w} \vee \bar{x})$

   Now find an assignment of variables such that maximum number of clauses is satisfied. Compute your solution using branch and bound algorithm. Write down a suitable bound function for the problem. Show detailed calculation steps through a branch and bound search tree.

8. (a) Define the following classes of problems: P, NP and NP-complete. **(8)**

   (b) What are the steps to prove a problem X is NP-Complete? Prove that decision version of independent set problem is NP-complete using reduction from 3-CNF satisfiability (3-CNF-SAT or 3-SAT) problem. **(3+12)**

   (c) Give a sketch (intuitive explanation) of the proof of Cook-Levin theorem which states that circuit-satisfiability (CIRCUIT-SAT) problem is NP-Complete. **(12)**

-----------------------------------------------

**L-2/T-1/CSE** **Date : 27/01/2016**

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-1 B. Sc. Engineering Examinations 2014-2015

Sub : **CSE 203** (Data Structures)

Full Marks: 210 Time : 3 Hours

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

---

## SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

1. (a) Prove that any comparison sort algorithm requires $\Omega(n \log n)$ comparisons in the worst case. **(10)**

(b) Write a linear-time Bucket Sort algorithm. Prove that time complexity of the algorithm is linear. **(10)**

(c) Write the Quick Sort algorithm. Analyze the best-case and worst-case time complexity of the algorithm. **(15)**

2. (a) Write an algorithm for building a heap. Show that an $n$-element heap can be built using $O(n)$ time. **(10)**

(b) What are the main operations of a priority queue? How could we implement these operations using a heap? **(10)**

(c) Compare linked lists and skip lists. Show that the expected space used by a skip list with $n$ entries is $O(n)$. **(15)**

Draw the skip-list for the following table that shows the keys and consecutive numbers of the heads found in toss while inserting the keys in the skip-list.

| Key | 15 | 40 | 35 | 28 | 21 | 65 | 84 | 79 | 50 |
|---|---|---|---|---|---|---|---|---|---|
| No. of consecutive heads | 1 | 4 | 1 | 0 | 3 | 2 | 2 | 3 | 1 |

3. (a) Explain, with examples, the single rotation and double rotation operations on AVL trees. **(10)**

(b) What is a Multiway Search tree? How do we search a key in a Multiway Search tree? **(10)**

(c) What makes a good hash function? Explain the three commonly used techniques to compute the probe sequences required for open addressing in a hash table. **(15)**

4. (a) Draw the binary search tree that results after successively inserting the keys 15, 8, 24, 12, 40, 18, 20, 23 into an initially empty binary search tree. Then, by using necessary rotations, redraw the binary search tree so that it becomes an AVL tree. **(10)**

Contd .......... P/2

## CSE 203

**Contd... Q. No. 4**

(b) Write the properties of a red-black tree. Show that the height of a red-black tree $T$ storing $n$ items is $O(\log n)$. **(10)**

(c) Explain, with examples, the four cases that may arise for deleting a node from a red-black tree. **(15)**

### SECTION – B

There are **FOUR** questions in this section. Answer any **THREE**.

5. (a) Suppose that you are given the following doubly linked list implementation that uses head and tail pointers: **(15)**

```
struct listNode
{
        int item ; //will store data
        struct listNode *next ; //will keep address of next node
        struct listNode *prev ; //will keep address of previous node
} ;
struct listNode * head ; //points to the first node of the list
struct listNode * tail ; //points to the last node of the list
```

Implement the function *"void delete(int N)"* that deletes all $M$th nodes of the list such that $M \bmod N = 0$ where $N > 0$. For example if $N = 3$, the function should delete 3rd node, 6th node, 9th node, and so on. The node pointed by *head* is the 1st node. Your implementation should be as efficient as possible.

(b) Compare the worst case run times of *ArrayList* and *LinkedList* (doubly with heat and tail pointers) for the following operations: **(4×3=12)**

(i) Insert an item at the $N$th position of the list (need to preserve order of items)

(ii) Remove the first item of the list (need to preserve order of item)

(iii) Remove the $N$th item of the list (need to preserve order of item)

(iv) Get (without removing) the $N$th item of the list

(c) Give two advantages and disadvantages of using array based lists over linked lists. **(8)**

6. (a) Suppose we create a binary search tree by inserting the following values in the given order: 50, 10, 13, 45, 55, 110, 5, 31, 64, and 47. Answer the following questions: **(5+4+4+6=19)**

(i) Draw the binary search tree.

(ii) Show the output values if we visit the tree using pre-order traversal technique.

(iii) Show the output values if we visit the tree using post-order traversal technique.

(iv) Show the resulting trees after we delete 47, 110, and 50. (Each deletion is applied on the original tree.)

Contd .......... P/3

## CSE 203

**Contd... Q. No. 6**

(b) Suppose that we are given the following definition of a rooted binary tree (**not binary search tree**): **(12)**

```
struct treeNode
{
        int item ; //stores the value
        struct treeNode * left ; //points to left child
        struct treeNode * right ; //points to right child
} ;
struct treeNode * root ; //global variable to store tree root node
```

Implement the following function *clone* (**must be recursive**) that makes a copy of a binary tree (keeping the original tree intact). The function will be called using the original root pointer as input argument. The function should return the root node of the new copy. For example calling $x = clone(root)$ will return the new root (of the tree copy) into variable $x$.

```
struct treeNode * clone(struct treeNode * node);
```

(c) Give two reasons why we should use restrictive data structures (e.g., Stack and Queue) instead of general purpose data structures. (e.g., List) **(4)**

7. (a) Suppose you are given the following definition of a singly linked list having head and tail pointers; **(15)**

```
struct listNode
{
        int item ; //will be used to store value
        struct listNode *next ; //will keep address of next node
} ;
struct listNode * head ; //points to first node of the list
struct listNode * tail ; //points to last node of the list
```

Implement a **stack** data structure using the above linked list definition. You need to implement only the following functions: *push, pop,* and *size*. The *size* function returns the current size (number of items) of the stack. You cannot use any other global or static variable in your implementation except *head* and *tail* pointers given above. Your implementations should be as efficient as possible.

(b) Suppose that we double the memory of an array based list every time we see that the memory is full during insertion of a new item. Show that the average time required per insertion to copy the existing items from the old memory to the new memory is constant. **(12)**

Contd .......... P/4

## CSE 203

**Contd... Q. No. 7**

(c) Give the idea of a data structure that can be used to implement a general tree where nodes can have more than two children. Re-draw the binary search tree shown in Fig. for Q. 7(c) using your proposed data structure. **(8)**

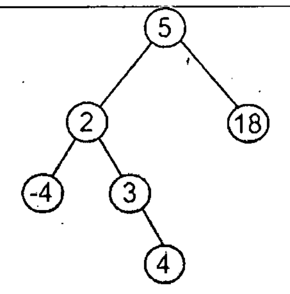



Fig. for Q. 7 (c)

8. (a) Consider the directed graph shown in Fig. for Q. 8(a). Answer the following questions: **(6+6=12)**

   (i) Perform depth first search on the graph starting from vertex $A$. Choose the smallest (in alphabetical order) vertex when there is a choice. Find the discovery time and finishing time of each vertex.

   (ii) Perform breadth first search on the graph starting from vertex $A$. Choose the smallest (in alphabetical order) vertex when there is a choice. Find the distance and parent of each vertex.

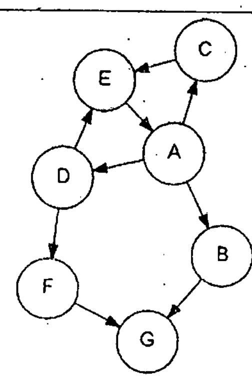



Fig. for Q. 8(a)

(b) For each of the following operations in an undirected graph, find the worst case running times in Big-**O** notation if we use an adjacency matrix; and if we use an adjacency list: **(4+4+6=14)**

   (i) Determine whether a given vertex is connected to no other vertex.

   (ii) Determine whether the graph has a vertex that is connected to no other vertex.

   (iii) Determine whether a given vertex is connected to every other vertex. Give the idea (in two to three sentences) of your algorithm for adjacency list.

(c) Recall that an undirected graph is "connected" if there is a path from any vertex to any other vertex. If an undirected graph is not connected, it has multiple connected components. A "connected component" consists of all the vertices reachable from a given vertex, and the edges incident on those vertices. Suggest an algorithm based on DFS that counts the number of connected components in a graph. Give only the idea or skeleton of the algorithm. **(9)**

**L-2/T-2/CSE** **Date : 01/07/2015**

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-2 B. Sc. Engineering Examinations 2013-2014

Sub : **CSE 207** (Algorithms)

Full Marks : 210 Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

---

## SECTION – A

There are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) Write down two algorithms for recognizing a leap year. Deduce their complexities. **(5+5)**

(b) Write down two versions of Binary search algorithm. Deduce average case complexity of both of them. **(5+5)**

(c) If an element appearing later in an array is smaller than one element appearing earlier then this is called an inversion. So in the array 2 5 3 1 there are (1 + 2 + 1 + 0 =4 inversions). Given an array A of integers using the idea of mergesort construct an *O(nlogn)* algorithm for determining number of inversions. **(15)**

2. (a) Write down two algorithms for constructing a max heap and deduce their complexities. Discuss how Carlsson reduces the leading coefficient in the complexity of the heapsort algorithm. **(10)**

(b) Write down a linear time algorithm for finding maximum sum of a consecutive subsequence of an array. **(15)**

(c) Write down traditional algorithms for finding minimum and maximum of an array, and a recursive algorithm. Compare their complexities. **(10)**

3. (a) Write down the pseudo code of Quicksort algorithm including partition. Deduce its average case complexity. Simulate partition algorithm on the array 6, 2, 7, 3, 8, 4, 9, 5, 10, 1, 11 using the first element as the partitioning element. Show whenever elements are swapped. **(10)**

(b) Given a directed graph with edge list and weight AB(5), BC(2), AC(4), AD(6), CD(3), DB(1), DE(6), EB(3), EF(3), FB(4) construct a mincost arborescence rooted at A. (edge CD(3) means edge is directed from C to D and has weight 3). **(10)**

(c) Find a minimum spanning tree using both Kruskal's algorithm and Prim's algorithm starting from A in the underlying graph of Question 3(b) with edge weights mentioned in parentheses. **(15)**

## CSE 207

4. (a) Given a set of jobs with start and finish times write down an algorithm so that maximum number of jobs can be processed. Show counterexamples for some natural greedy algorithm that will not find optimal solutions. **(10)**

   (b) Solve the Multi_Peg Tower of Hanoi problem with (n,p)=(377,8). Find kmax, Na and $n_l$ for at least two subproblems. **(10)**

   (c) Given the message 'abaacbdbebfbcdcecf' construct a dynamic Huffman frees showing how trees are updated after sending each symbol and code of the symbol. **(15)**

### SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE**.

5. (a) Write a linear-time algorithm and an exponential-time algorithm for finding the *n*th Fibonacci number. Analyze the time-complexity of each algorithm. **(10)**

   (b) Write a parallel algorithm for computing the sum of *n* numbers. Also, write a parallel algorithm to multiply two $n \times n$ matrices that runs in $O(\log n)$ time. **(10)**

   (c) Using dynamic programming compute the minimum number of scalar multiplications for the matrix chain multiplication of $A_1$, $A_2$, $A_3$, $A_4$, $A_5$, $A_6$ with dimensions $15 \times 5$, $5 \times 50$, $50 \times 20$, $20 \times 10$, $10 \times 35$, $35\times 25$, respectively. Also, show the ordering of the matrices for the desired minimum number of scalar multiplications. **(15)**

6. (a) Compare dynamic programming and memorization techniques. **(10)**

   "A dynamic programming algorithm usually outperforms the corresponding memorized algorithm" – explain.

   (b) Write a memorized algorithm for solving the matrix chain multiplication problem. Analyze the time-complexity of the algorithm. **(10)**

   (c) What is a state-space tree? **(15)**

   Solve the following instance of the 0/1 knapsack problem using the branch-and-bound approach with a state-space tree. Assume that the knapsack capacity is 10.

| **Item** | **Weight** | **Value** |
|---|---|---|
| 1 | 7 | 42 |
| 2 | 4 | 40 |
| 3 | 3 | 12 |
| 4 | 5 | 25 |

7. (a) Write and prove the max-flow min-cut theorem. Explain how one can convert a multiple-source, multiple-sink maximum-flow problem into a problem with a single source and a single sink. **(10)**

   (b) Prove that the longest common subsequence problem has both the Overlapping Subproblems property and the Optimal Substructure property. **(10)**

Contd .......... P/3

**CSE 207**
**Contd ... Q. No. 7**

(c) Find all longest common subsequences of $X = \langle \mathbf{ATGCTGAT} \rangle$ and $Y = \langle \mathbf{TGGCATA} \rangle$ using dynamic programming method. **(15)**

8. (a) Write the names of four NP-complete problems. Define the following six classes of problems: P, Co-P, NP, Co-NP, NP- complete and NP-hard. By a diagram show the relationship among these classes of problems that most researchers regard as the most likely. **(10)**

(b) If $L_1$ and $L_2$ are two languages such that $L_1 \leq_p L_2$, then show that $L_2 \in$ P implies $L_1 \in$ P. If an NP-complete problem is polynomial solvable, then prove that P = NP. **(10)**

(c) Write a polynomial-time 2-approximation algorithm for the traveling-salesman problem where the edge weights satisfy the triangle inequality, and prove the approximation ratio of the algorithm.

If P $\neq$ NP, then prove that for any constant $\rho \geq 1$, there is no polynomial-time approximation algorithm with ratio $\rho$ for the general traveling-salesman problem. **(15)**

-----------------------------------

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-1 B. Sc. Engineering Examinations 2013-2014

Sub : **CSE 203** (Data Structures)

Full Marks: 210 Time : 3 Hours

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

---

## SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

1. (a) Write a sorting algorithm that can always sort $n$ elements in $O(n \log n)$ time. Analyze the best-case and worst-case time complexity of the algorithm. **(7+8=15)**

(b) Prove that any comparison sort algorithm requires $\Omega(n \log n)$ comparisons in the worst case. **(8)**

(c) Write the counting sort algorithm. Prove that the algorithm runs in linear time. Explain why counting sort is stable. **(3×4=12)**

2. (a) Write algorithms for the search, insertion and deletion operations of skip lists. Show that the expected search, insertion and deletion time is $O(\log n)$ in a skip list with $n$ entries. Also show that the expected space used is $O(n)$. **(18)**

(b) Draw the 11-item hash table that results from using the hash function $h(i) = (2i+5)$ mod 11, to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, assuming collisions are handled by double hashing using a secondary hash function $h'(k) = 7 - (k \bmod 7)$. **(12)**

(c) Compare linked lists and skip lists. **(5)**

3. (a) What is an AVL tree? Show that the height of an AVL tree $T$ storing $n$ items is $O(\log n)$. **(10)**

(b) Explain with examples the single rotation and double rotation operations on AVL tree. **(10)**

(c) Write the differences between a binary-search tree and a min-heap. **(4+6+5=15)**

Write two procedures for finding the tree-successor and the tree-predecessor in a binary-search tree.

For the set of {11, 4, 25, 10, 36, 17, 21} of keys, draw a binary search tree of height 3.

4. (a) Write the properties of a red-black tree. Show the red-black trees that result after successively inserting the keys 51, 46, 41, 22, 29, 18 into an initially empty red-black tree. **(15)**

(b) Explain the divide-and-conquer technique. Write the quick sort algorithm, and analyze the time-complexity of the algorithm. **(15)**

(c) Write the properties of a B-tree. **(5)**

17

Contd ......... P/2

**CSE 203/CSE**

**SECTION – B**

There are **FOUR** questions in this section. Answer any **THREE**.

5. (a) What is a data structure? When is a data structure called a linear data structure? Write 4 (four) properties of a linear data structure. **(1.5+1.5+4=7)**

(b) Let **a** : $array[l_1..u_1, l_2..u_2]$ be a 2D integer array. Assume b is the base address of the array a. Then write the Array Mapping Function to calculate **address(a [i, j])** in **(5+5=10)**

(i) Column-major order

(ii) Row-major order

(c) Given a doubly linked list and an integer n, write a function **deleteLastNElements()** that deletes last n nodes from the given linked list. **(8)**

(d) What is a proper binary tree? Construct **the proper binary tree** whose inorder and postorder traversals are given as: **(2+8=10)**

**Inorder** : BUETCSE12

**Postorder** : BEUCTES21

6. (a) How do you implement a Queue using a Stack? What are the running time of **enqueue** and **dequeue** operations of such implementation? **(8+2=10)**

(b) Which type of traversal will traverse the binary tree in figure for Question 6(b) as: **BUETCSE12**? Write an algorithm for such traversal of a binary tree using a Stack or Queue. **(2+10=12)**

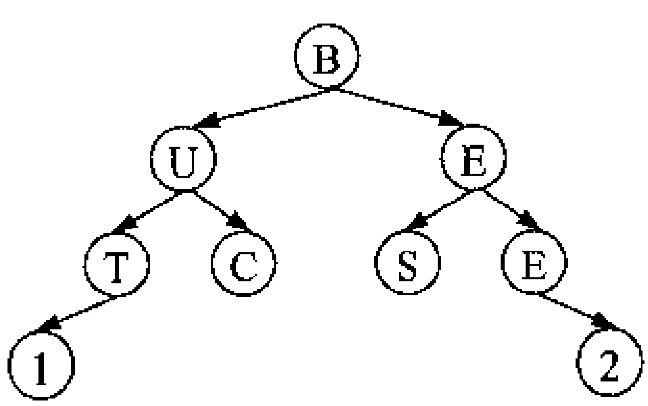



Figure for Question 6(b)

(c) Give a brief description of **(1.5+1.5+1.5+2.5=7)**

(i) Graph

(ii) Forest

(iii) Tree

(iv) Spanning subgraph

(d) Give a comparison of adjacency matrix and adjacency list representation of a graph. **(6)**

7. (a) Give 2 (two) real life applications each for **(2+2+2=6)**

   (i) Stack

   (ii) Queue

   (iii) Priority Queue

   (b) Write pseudocodes for **enqueue** and **dequeue** operations for the array implementations of a circular queue. **(7)**

   (c) Assume an object which has two elements {key, element}. An array of such object is given in figure for Question 7(c). Then **(8+8+6=22)**

   (i) Convert the array into Max-heap.

   (ii) Insert (36, B) in the resultant Max-heat of 7(c)(i)

   (iii) Show each step of the Heap sort over the resultant Max-heap of 7(c)(ii).

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14, U | 11, E | 13, T | 12, C | 26; S | 19, E | 20, W | 24, P | 18, O | 17, L | 7, A | 8, S | 9, H | 10, I |

Figure for Question 7(c)

8. (a) Give a comparison of array and linked list. **(6)**

   (b) Draw a binary tree whose preorder and inorder traversals both yield **BUETCSE12**. **(7)**

   (c) Consider a graph G as shown in figure for Question 8(c). Then **(6+6+4+6=22)**

   (i) Write the adjacency matrix and adjacency list of the Graph G.

   (ii) Draw the BFS forest of the graph G by assuming B as the starting vertex.

   (iii) Write the sequences of vertices if you explore the graph G starting from vertex B in DFS.

   (iv) Write the tree edges, back edges, forward edges and cross edges.

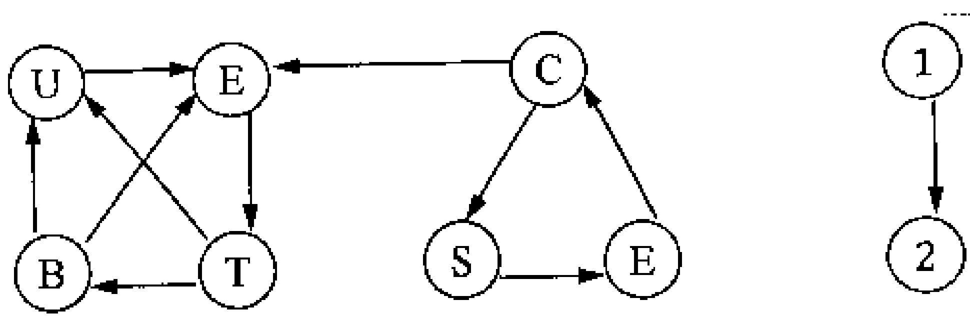



Figure for Question 8(c)

---

L-2/T-2/CSE **Date : 08/12/2014**

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-2 B. Sc. Engineering Examinations 2012-2013

Sub : **CSE 207** (Algorithms)

Full Marks : 210 Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

## SECTION – A

There are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) Does the Dijkstra's algorithm terminate, if it is applied on a graph $G$ that contains a negative-weight cycle? Why or why not? Justify your answer. **(5)**

(b) What does $d[v]$ represent at the end of the execution of Algorithm 1, if the algorithm is applied on an unweighted directed acyclic graph (DAG) $G$ with a source s? What is the runtime complexity in terms of $|V|$ and $|E|$ of Algorithm 1? **(6+6)**

**Algorithm 1**

```
compute a topological sorting of G
for each vertex v in V[G]
    d[v] ← 0
for each u taken in topologically sorted order
    for each v in adj(u)
        if d[v] < d[u] + 1
            d[v] ← d[u] + 1
```

(c) In Figure 1, the shortest path estimate of each vertex appears within the vertex and the distance to traverse from a vertex to another appears as the weight of the corresponding directed edge. Show the updated shortest path estimates of vertices after relaxing the following edges. **(6)**

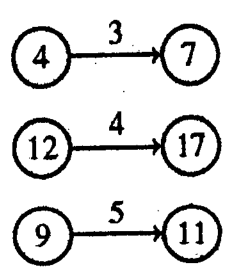



Figure 1

(d) State and prove the convergence property of the shortest paths. **(12)**

2. (a) Write the Bellman-Ford algorithm. Analyze the time complexity and correctness of the Bellman-Ford algorithm. **(20)**

(b) Apply the Transitive-Closure algorithm on the graph given in Figure 2. Show the matrices computed at all stages of the execution of the Transitive-Closure algorithm. **(15)**

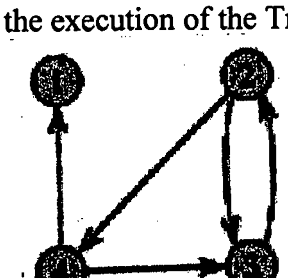


Figure 2

Contd .......... P/2

## CSE 207

3. (a) State the parenthesis theorem for the depth-first search. **(6)**

(b) Assume that a breath-first tree has already been computed with BFS on a graph $G$ with the source vertex $s$. Write an algorithm to point out the vertices on a shortest path from $s$ to $v$. **(8)**

(c) Prove that the component graph is a DAG. **(8)**

(d) Write a linear time algorithm to compute the strongly connected components of a directed graph $G = (V, E)$ using two depth-first searches. Give an example to show every step of the algorithm. **(8+5)**

4. (a) Explain the asymptotic upper, lower and tight bounds of an algorithm. **(9)**

(b) Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n/4) + T(n/2) + n^2$. **(10)**

(c) If prim's algorithm is applied on the graph in Fig. 3, which will be the fifth edge to be added onto the spanning tree being formed. Start from vertex "a". **(10)**

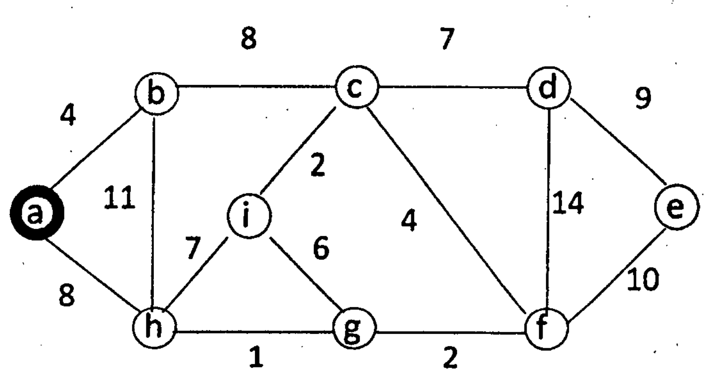



Figure 3

(d) Write pesudocode of FIND-SET algorithm that works using the path compression heuristic. FIND-SET($x$) procedure returns a pointer to the representative of the set containing $x$. **(6)**

### SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE**.

5. (a) Describe the pseudocode of a *divide and conquer* sorting algorithm that runs in $O(n^2)$ time in the worst case. **(15)**

(b) Deduce the best case running time of your algorithm in Q. 5(a). **(7)**

(c) An inversion in an array $A[1..n]$ is a pair of indices $(i, j)$ such that $i < j$ and $A[i] > A[j]$. The number of inversions in an $n$-element array is between 0 (if the array is sorted) and $\frac{n}{2}$ (if the array is sorted backward). Describe and analyze an algorithm to count the number of inversions in an $n$-element array in $O(nlogn)$ time (hints: modify *mergesort* algorithm). **(13)**

Contd .......... P/3

## CSE 207

6. (a) Suppose we are given a sequence of integers $A[1..n]$ and we want to find the longest subsequence whose elements are in decreasing order. More concretely, we want to find the longest sequence of indices $1 < i_1 < i_2 < \dots i_k < n$ such that $A[i_j] > A[i_{j+1}]$ for all $j$. Describe the pseudocode of a backtrack algorithm that solves this problem is $O(2^n)$ time. **(15)**

   (b) In a previous life, you worked as a cashier in the ancient city of Egypt, spending the better part of your day giving change to your customers. Because paper was a very rare and valuable resource at the time, cashiers were required by law to use the fewest notes possible whenever they gave change. The currency of Egypt at that time, called Dream Dollars, was available in the following denominations: \$1, \$4, \$7, \$13, \$28, \$52, \$91, \$ 365. **(20)**

   (i) A greedy change algorithm repeatedly takes the largest note that does not exceed the target amount. For example, to make \$122 using the greedy algorithm, we first take a \$91 note, then a \$28 note, and finally three \$1 notes. Give an example where this greedy algorithm uses more Dream Dollar notes than the minimum possible.

   (ii) Describe a dynamic programming algorithm that computes, given an integer k. the minimum number of notes needed to make k Dream Dollars. Write down the pseudocodes for both memorized approach and bottom-up interactive approach. Analyze the running-times of both approaches.

7. (a) Suppose you are given n tasks with start times $S[1..n)$ and finish times $F[1.n]$. Your job is to schedule as many tasks as possible (in one room) so that no two scheduled tasks overlap. We say that two tasks I and j do not overlap if $S[i] \geq F[j]$ or $S[j] \geq F[i]$. **(15)**

   (i) Describe a greedy algorithm that solves the above problem i.e., finds the maximum number of non-overlapping tasks.

   (ii) Prove the correctness of your greedy algorithm.

   (b) Show that the decision version of vertex cover optimization problem is *NP-Complete* using a reduction from Independent Set decision problem. **(12)**

   (c) Prove that the optimization version of Independent set problem is *NP- Hard.* **(8)**

8. (a) On an auspicious day, while walking alone in a forest, assume that you find $n$ magic boxes kept in a row. Each box is numbered from 1 to n sequentially and contains some dollars! While you are about to touch the first box, a giant Genie appears and told you that you can open a box and take the dollars in it. However, you can only open a box (say box $i$) provided that you did not open the previous two boxes (box $i-1$ and box $i-2$). Given the amount of dollars kept in each box D[1..n], your task is to find the maximum number of dollars that you can obtain from the $n$ boxes. Describe a dynamic programming solution to solve this problem. Analyze the running time of your solution. **(12)**

**Contd ... Q. No. 8**

(b) Consider the following greedy approximation algorithm for the vertex cover optimization problem. **(15)**

```
Algorithm GreedyVertexCover(G)
C ← ϕ
while E is not empty              //E is the set of edges
      u ← vertex in V with maximum degree        //V is the set of
                                                 //vertices
      V ← V − {u}
      C ← C ∪ {u}
      for each edge (u,v) adjacent to u
            E ← E − {(u,v)}          //remove all edges from E that are
                                     //covered by vertex u
return C      //C is the vertex cover produced by this algorithm
```

(i) Give an example graph where the above algorithm may not produce the optimal vertex cover.

(ii) Show that the approximation ratio of the above greedy algorithm is $O(\lceil \lg |E| \rceil)$.

(c) Give an efficient 2-approximation algorithm for vertex cover optimization problem. **(8)**

---

L-2/T-1/CSE Date : 12/05/2014

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-1 B. Sc. Engineering Examinations 2012-2013

Sub : **CSE 203** (Data Structures)

Full Marks: 210 Time : 3 Hours

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

---

## SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

1. (a) What is a data structure? Write the properties of a linear data structure. **(1+4=5)**

   (b) Prove that $5n^3 + 3n = O(n^4)$, but $5n^3 + 3n \neq O(n^2)$. **(5+5=10)**

   (c) Let a : $array[l_1..u_1, l_2..u_2, l_3..u_3]$ be a 3-dimensional array. Assume that $b$ and $c$ are the base address and component length of the array a, respectively. Then write the Array Mapping Function to calculate addr(a[$i$, $j$, $k$]). **(10)**

   (d) Write the advantages and disadvantages of arrays and linked-lists. **(10)**

2. (a) Write the differences between a standard queue and a priority queue. Write the property of a Max-Heap. Write an algorithm for building a Max-Heap. Analyze the time-complexity of the algorithm. **(3+3+7+7=20)**

   (b) Explain the divide-and-conquer technique? What is the function of the pivot element in the quick sort algorithm? **(3+2=5)**

   (c) Write two functions for the insertion and deletion operations in a circular linked list. **(5+5=10)**

3. (a) Write an algorithm for DFS traversal of a given graph. Analyze the time-complexity of the algorithm. **(8+7=15)**

   (b) Consider a graph $G$ as shown in Figure for Question 3(b). Then **(6+4+4+6=20)**

   (i) Write the adjacency matrix and adjacency list of the graph $G$.

   (ii) Draw the BFS forest of the graph $G$ by assuming $b$ as the starting vertex.

   (iii) Write the sequence of vertices if you explore the graph $G$ starting from vertex $b$ in DFS.

   (iv) Write the tree edges, back edges, forward edges, and cross edges.

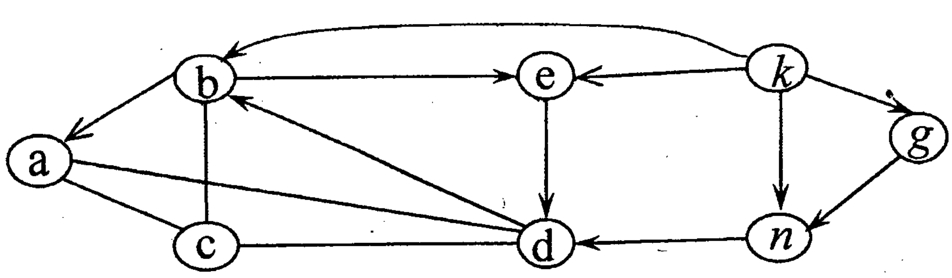



**Figure for Question 3(b)**

Contd .......... P/2

**CSE 203**

4. (a) What is a tree? Write a linear-time algorithm for finding the height of a tree. Analyze the time-complexity of the algorithm. **(1+4+4=9)**

(b) Let $T$ be a proper binary tree with height $h$. Then, prove that the number of internal nodes in $T$ is at least $h$ and at most $2^h - 1$. **(4+4=8)**

(c) Construct the binary tree whose inorder and postorder traversals are given as: **(10)**

Inorder : $i\,f\,d\,j\,g\,c\,a\,b\,k\,h\,m\,e$

Postorder : $i\,j\,g\,d\,f\,k\,m\,h\,e\,b\,a\,c$

(d) Draw the Linked Structure for the binary tree of Question 4(c). **(8)**

**SECTION – B**

There are **FOUR** questions in this section. Answer any **THREE**.

5. (a) Sort the following dataset using "quicksort" in descending order when pivot element is the first element. **(12)**

$D = \{6, 0, 2, 0, 1, 3, 6, 1\}$

(show the steps)

(b) If you use "merge-sort", what will be the merge-tree for the data set mentioned in Q. 5(a)? Show that merge sort runs in time $\theta$ (*nlgn*) in the wrost-case. **(6+4=10)**

(c) Analyze the average case running time for "bucket sort". **(8)**

(d) Write the "counting sort" steps for the dataset of Q. 5(a). **(5)**

6. (a) Write pseudo-code for finding a key, inserting a key and deleting a key in a hash table using quadratic probing. Explain your method with an example. **(12)**

(b) Prove that in a hash table in which collisions are resolved by "chaining", a successful search takes average case time $\theta(1+\alpha)$, under the assumption of uniform hashing, where $\alpha$ is the load factor. **(10)**

(c) Insert key in a hashtable of size 13 using double hashing with **(13)**

$h_1(k) = k$ mod hash-table size

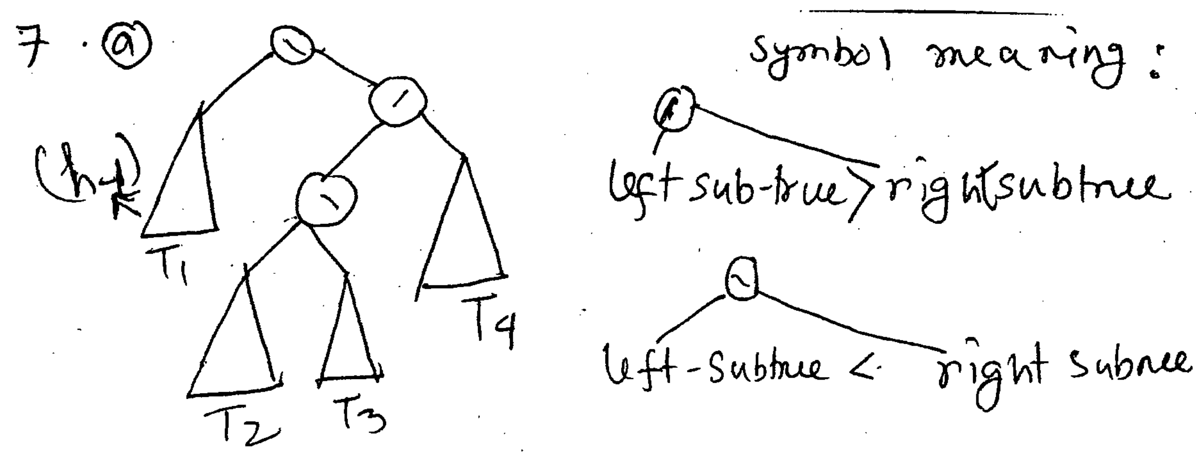
and $h_2(k) = 8 - (k \text{ mat } 8)$,

where $k$ is the key, $h_1$ and $h_2$ are hash functions. Key insertion order is–

18, 41, 22, 44, 59, 32, 31, 73.

## CSE 203

7\. (a) **(4+4+7=15)**




If $T_1$ has height h-1, what will be the heights of $T_2$, $T_3$, and $T_4$? What will happen if you delete an element from $T_1$? Discuss different cases (all possible) for deletion operation in an "AVL tree".

(b) Design an AVL tree with keys 5, 12, 3, –2, –5, –3, 1. Show the steps. **(15)**

(c) What is the height for AVL tree? (Assume that the number of internal nodes of the AVL tree is *n*.) **(5)**

8\. (a) Write down the properties of a "red-black tree". With an example show insertion and deletion operations and their costs of a "red-black" tree. **(5+15=20)**

(b) Build "2-3-4 tree" with the keys 12, 5, 6, 10, 15, 8, 13, 17, 11, 14, 4. Delete elements in the following sequence: 4, 12, 13. (show the steps). **(15)**

---

L-2/T-2/CSE **Date : 06/07/2013**

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-2 B. Sc. Engineering Examinations 2011-2012

Sub : **CSE 207** (Algorithms)

Full Marks: 210 Time : 3 Hours

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

---

## SECTION – A

There are **NINE** questions in this section. Answer any **SEVEN**.

1. "Search spaces for natural combinatorial problems tend to grow exponentially with the size of the input" — explain and justify with necessary examples. **(15)**

2. Prove that greedy method stays ahead in interval scheduling problem. And hence show that greedy method returns an optimal set of jobs. **(15)**

3. Prove that if we use greedy algorithm for interval partitioning problem, every interval will be assigned a label, and no two overlapping intervals will receive the same label. **(15)**

4. Explain how Dijkstras algorithm can be implemented using priority queue. Find out the running time for this algorithm. **(15)**

5. Put down and prove the "cut property" and "cycle property". Explain with necessary examples how these properties are useful in solving the minimum spanning tree problem. **(15)**

6. Describe with necessary elaboration on the required operations and data structure, how the union-find data structure can be used for Kruskal's algorithm. **(15)**

7. We have Union-Find data structure for some set $S$ of size $n$, where unions keep the name of the larger set. Prove that for array implementation any sequence of $k$ Union operations takes at most $O(k \log k)$ time. Also prove that for pointer-based implementation, a Find operation takes $O(\log n)$ time. **(15)**

8. For a ranking of "7, 9, 1, 5, 4", discuss how the algorithm for finding inversion counts using the divide and conquer approach will work. Show the various procedure calls and returned values using a neat schematic diagram. In a division process, you can put $\lceil n/2 \rceil$ elements in the left half, and $\lfloor n/2 \rfloor$ elements in the right half, where $n$ is the number of elements being divided. Then find out the running time of this algorithm. **(15)**

9. In the finding the closest pair of points problem using the divide and conquer approach, if $q$ is a point in the left half and $r$ is a point in the right half, and $d(q,r) < \delta$, where $\delta$ is the minimum of distances in each half, then show that each of $q$ and $r$ lies within a distance $\delta$ of the line dividing the two halves. Hence show that $q$ and $r$ are within 11 positions of each other in a list sorted in $y$ coordinate. **(15)**

Contd .......... P/2

## CSE 207

### SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE**.

10. (a) Compare the properties of problems that can be solved with greedy method, divide-and-conquer method and dynamic programming method. **(8)**

(b) Write and prove the optimal-substructure property of the matrix chain multiplication problem. Calculate the number of parenthesizations of a sequence of $n$ matrices. **(5+5=10)**

(c) Let $X = \langle x_1, x_2, \dots x_m \rangle$ and $Y = \langle y_1, y_2, \dots y_n \rangle$ be sequences, and let $Z = \langle z_1, z_2, \dots z_k \rangle$ be any LCS of $X$ and $Y$. If $x_m \neq y_n$ and $z_k \neq x_m$, then prove that Z is an LCS of $X_{m-1}$ and $Y$. **(7)**

(d) Find an LCS of $X = \langle \text{ATCTGAT} \rangle$ and $Y = \langle \text{TGCATA} \rangle$ by showing the $c$ and $b$ tables. **(10)**

11. (a) Write and prove the max-flow min-cut theorem. **(10)**

(b) Explain why one can solve the fractional knapsack problem by a greedy strategy, but one cannot solve the 0-1 knapsack problem by such a strategy. **(10)**

(c) Find an optimal solution to the 0/1 knapsack instance of $n = 4$, $W = 6$, $(v_1, v_2, v_3, v_4) = (50, 30, 12, 45)$, and $(w_1, w_2, w_3, w_4) = (2, 3, 1, 3)$. **(15)**

12. (a) What are the traveling-salesman problem, Euler tour problem and Hamiltonian cycle problem? If $L_1 \leq_p L_2$ and $L_2 \leq_p L_3$, then prove that $L_1 \leq_p L_3$. **(3+7=10)**

(b) What is a flow network? What is the maximum-flow problem? Explain how one can convert a multiple-source, multiple-sink maximum-flow problem into a problem with a single source and a single sink. **(4+2+4=10)**

(c) Write the Ford-Fulkerson algorithm for solving the maximum-flow problem, and analyze the time-complexity of the algorithm. Explain how the Edmonds-Karp algorithm improves the bound of the Ford-Fulkerson algorithm. **(10+5=15)**

13. (a) Define the following six classes of problems: P, Co-P, NP, Co-NP, NPC and NP-hard. By a diagram show the relationship among P, Co-P, NP, Co-NP and NPC that most researchers regard as the most likely. **(6+4=10)**

(b) If $L_1$ and $L_2$ are two languages such that $L_1 \leq_p L_2$, then show that $L_2 \in P$ implies $L_1 \in P$. If any NP-complete problem is polynomial-time solvable, then prove that P = NP. **(7+3=10)**

(c) Write a polynomial-time 2-approximation algorithm for the vertex-cover problem, and prove the approximation ratio of the algorithm. If $P \neq NP$, then prove that for any constant $\rho \geq 1$, there is no polynomial-time approximation algorithm with ratio $\rho$ for the general traveling-salesman problem. **(8+7=15)**

---

L-2/T-1/CSE Date : 17/12/2012

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-2/T-1 B. Sc. Engineering Examinations 2011-2012

Sub : **CSE 203** (Data Structures)

Full Marks : 210 Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

---

## SECTION – A

There are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) Prove that $O(n^3)\,O(n) = O(n^4)$. **(5)**

(b) Write the main three reasons why we usually concentrate on finding the worst-case running time of an algorithm. **(5)**

(c) Write an algorithm for DFS traversal of a given graph. Analyze the time-complexity of the algorithm. **(10+5=15)**

(d) Write the sequences of the vertices if you explore the following graph G using BFS and DFS. Classify the edges of G for your DFS traversal. **(4+6=10)**

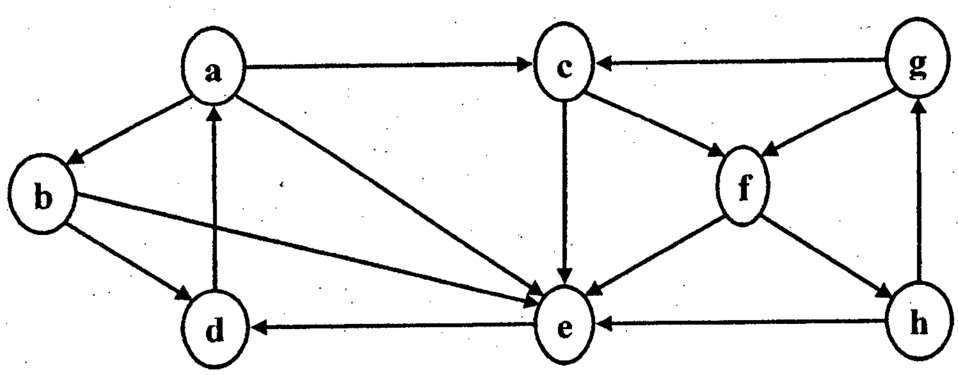



**Figure for Question 1(d)**

2. (a) What is the divide-and-conquer technique? **(5)**

(b) Write the merge sort algorithm. Analyze the time-complexity of the algorithm. **(10+5=15)**

(c) Write the recurrence relation of the merge sort algorithm. Draw the recursion tree for the recurrence relation of the merge sort algorithm. **(7)**

(d) Sort the array 15, 13, 9, 5, 12, 8, 7, 4, 10, 6, 2, 1, 26, 23 using heap sort showing the values in the right of the nodes. **(8)**

3. (a) Write the quick sort algorithm. Simulate the quick sort algorithm on the array of Question 2(d) showing in detail how partition routine works. **(8+7=15)**

(b) What is a priority queue? Write three applications of a priority queue. **(2+3=5)**

(c) What are the main operations of a priority queue? Explain how one can implement these operations using a heap. **(15)**

Contd .......... P/2

## CSE 203

4. (a) Explain the advantages and disadvantages of arrays and linked-lists. **(5)**

   (b) What is a tree? Write an algorithm for finding the height of a tree. Analyze the time-complexity of the algorithm. **(2+5+5=12)**

   (c) Let T be a binary tree with height h. Then prove that the number of nodes in T is at least $2h + 1$ and at most $2^{h+1} - 1$. **(12)**

   (d) Write pseudocodes for Insert and Delete operations of a circular queue. **(6)**

## SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE.**

5. (a) Explain the insertion and deletion operations in a binary search tree with illustrative examples. **(14)**

   (b) Sort the following data set using counting sort showing every step of the algorithm. **(11)**

   $A = \{6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2\}$

   (c) "An important property of counting sort is that it is stable" – explain. **(4)**

   (d) Analyze the average case running time for bucket sort. **(6)**

6. (a) Draw the skip-list for the following table that shows the keys and consecutive numbers of the heads found in toss while inserting the keys in a probabilistic skip-list. **(7+8=15)**

| key | 3 | 6 | 7 | 9 | 12 | 17 | 19 | 21 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|
| No. of consecutive heads | 1 | 4 | 1 | 2 | 1 | 2 | 1 | 1 | 3 | 1 |

   Do the following operations on the skip-list and show the procedures.

   (i) Search 21

   (ii) Insert 22 with 3 consecutive heads

   (iii) Delete 19

   (iv) Search 18

   (b) Show that the expected running time of searching in a probabilistic skip-list is O (log n). **(6)**

   (c) Consider inserting the keys 18, 41, 22, 44, 59, 32, 31, 73 into a hash table of length $m = 13$ using double hashing with $h_1(k) = k \bmod 13$,

   $$h_2(k) = 8 - (k \bmod 8).$$

   Draw the hash table. **(10)**

   (d) What are the properties of a good hash function? **(4)**

# CSE 317

**Contd ... Q. No. 6**

(c) Solve the following problem using Third-Order RK method over the interval from $x = 0$ to $x = 1$ with a step size of $h = 0.5$ where $y(0) = 1$. **(11)**

$$y' = x^3 y - 1.5y$$

Use the following version of Third-Order RK method.

$$y_{i+1} = y_i + h\,(k_1 + 4k_2 + k_3)/6$$

where,

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + h/2,\; y_i + k_1 h/2)$$

$$k_3 = f(x_i + h,\; y_i - k_1 h + 2k_2 h)$$

7. (a) Derive the integral of the following tabular data using best combination of Trapezoidal rule and Simpson's rule. **(10)**

| x | 1 | 2 | 3 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| f(x) | −1 | 1 | 3 | 7 | 9 | 11 | 13 |

(b) Fit the function $f(x; a, b) = a\,(1 - e^{-bx})$ to the data: **(25)**

| x | 0.25 | 0.75 | 1.25 | 1.75 | 2.25 |
|---|---|---|---|---|---|
| y | 0.28 | 0.57 | 0.68 | 0.74 | 0.79 |

Use initial guesses of $a = 1.0$ and $b = 1.0$ for the parameters.

8. (a) Use Romberg integration with an accuracy of $O(h^8)$ to integrate the following function from $x = 0$ to $x = 0.8$. **(15)**

$$f(x) = 7x^3 + 4x^2 + 5x + 3$$

(b) Integrate $y' = 4e^{0.8x} - 0.5y$ from $x = 0$ to $x = 1$ with a step size of 0.5 using the following methods: **(20)**

(i) Heun's method

(ii) Midpoint method

Given, $y(0) = 2$.

---

**CSE 481**

8. (a) Explain why **Si** is used for wafer processing. **(5)**

(b) Describe the effect of **collector resistance** $R_c$ on the propagation delay of Bi-CMOS. **(5)**

(c) Explain why Bi-CMOS is faster than CMOS for high capacitive load, but slower for low capacitive load. **(15)**

(d) Show that, for clocked-CMOS, **charge leakage** places a lower limit on allowable clock frequency and it is **(10)**

$$f = (1/t_h) \text{ where } t_h = (C_{out}/I_L)(V_1 - V_x)$$

---