|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Unit – I** | | | **Marks** | **CO** | **PO** | **BTL** |
| 1. | A) | List the areas where data structures can be applied. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Describe node structure of Single linked list and double linked list. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Develop algorithms to create and insert elements into a single linked list. | (10 Marks) | C202.1 | PO1,PO2 | 3 |
|  | | | | | |
| 2. | A) | Define data structure. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Discuss which data structure used in recursion. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Demonstrate stack operations using linked list. | (10 Marks) | C202.1 | PO1,2 | 3 |
|  | | | | | | |
| 3. | A) | List the applications of stack. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Discuss queue operations using arrays. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Explain the representation of doubly linked list and also explain Deletion, Search and Traversing operations on singly linked list. | (10 Marks) | C202.1 | PO1,2 | 3 |
|  | | | | | | |
| 4. | A) | Describe stack overflow and underflow conditions when it is implemented by using arrays and linked list. | (2 Marks) | C202.1 | PO1,PO2 | 1 |
| B) | Distinguish linear and non-linear data structure. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Discuss about the various representations of a queue and its operations. | (10 Marks) | C202.1 | PO1,2 | 3 |
|  | | | | | | |
| 5. | A) | Identify an element is existed in a singly linked list with an algorithm. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Discuss the differences between array and linked list. | (3 Marks) | C202.1 | PO1,2 | 2 |
| C) | Explain the representation of circular linked list with create and display operations. | (10 Marks) | C202.1 | PO1,2 | 3 |
|  | | | | | | |
| 6. | A) | Descrie an algorithm to delete the first element from doubly linked list. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Explain how to display node values in singly linked list. | (3 Marks) | C202.1 | PO1,2 | 2 |
| C) | Explain the following   1. (A+B)\*C convert Infix to postfix Expression by using stack 2. 456\*+ Evaluate this expression using stack | (10Marks) | C202.1 | PO1,2 | 3 |
| 7 |  | List any three differences between stacks and queue. | (2 Marks) | C202.1 | PO1 | 1 |
|  | Discuss an algorithm to display the content of linked list in reverse order. | (3 Marks) | C202.1 | PO1,2 | 2 |
|  | Develop a C program to convert infix expression to postfix expression. | (10 Marks) | C202.1 | PO1,2 | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Unit – II** | **Marks** | **CO** | **PO** | **BTL** |
| 1. | A) | Define Dictionary with an example. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Describe hashing and state the use of hashing. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Write linear list representation of dictionaries and it’s insertion operation with an example. | (10 Marks) | C202.1 | PO1,2 | 3 |
|  |  |  |  |  |  |
| 2. | A) | Identify various applications of dictionaries | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Describe hash function along with properties of hash function. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Demonstrate the following  a)Rehashing b) Extendible hashing. | (10 Marks) | C202.1 & C202.2 | PO1,2 | 3 |
|  |  |  |  |  |  |
| 3. | A) | Define mid square method of hashing. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Explain algorithm to perform retrieve operation on dictionaries. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Demonstrate linear probing. Insert pairs whose keys in order are 7, 42, 25, 70, 14, 38, 8, 21, 34, 11 into a hash table with b = 13 buckets using the hash function f(k) = k mod b. start with an empty hash table and draw the hash table following each insert? | (10 Marks) | C202.1 & C202.2 | PO1,2 | 3 |
|  |  |  |  |  |  |
| 4. | A) | Define digit folding method of hashing. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Explain Skip list representations. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Demonstrate quadratic probing and insert pairs whose keys in order are 37, 90, 55, 22, 17, 49, 87 into a hash table of size 10. | (10 Marks) | C202.1 & C202.2 | PO1,2 | 3 |
|  |  |  |  |  |  |
| 5. | A) | Define hash clashing or collision. State different types of collision resolution techniques. | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Explain multiplicative method of hashing. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Illustrate double hashing and Insert pairs whose keys in order are 37, 90, 55, 22, 17, 49, 82 into a hash table of size 10. | (10 Marks) | C202.1 & C202.2 | PO1,2 | 3 |
|  |  |  |  |  |  |
| 6. | A) | Define hash table and hash function? | (2 Marks) | C202.1 | PO1 | 1 |
| B) | Explain division method of hashing. | (3 Marks) | C202.1 | PO1 | 2 |
| C) | Illustrate chaining and Insert pairs whose keys in order are 7, 24, 18, 52, 36, 54, 11, and 23 into a chained hash table of 9 memory locations.  Use h(k) = 2k+3 mod m. | (10 Marks) | C202.1 & C202.2 | PO1,2 | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Unit – III** | **Marks** | **CO** | **PO** | **BTL** |
| 1. | A) | Define Binary Tree with example | (2 Marks) | C202.3 | PO1 | 1 |
| B) | Explain Properties of Binary Tree | (3 Marks) | C202.3 | PO1 | 2 |
| C) | Prepare binary search tree for the following data: 56,32,11,75,29,85,46,88,22,5,38,14,72,9,66 | (10 Marks) | C202.3 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 2. | A) | Difference between Predecessor and Successor | (2 Marks) | C202.3 | PO1 | 1 |
| B) | Explain various balanced search trees | (3 Marks) | C202.3 | PO1 | 2 |
| C) | Demonstrate importance of Red Black Trees and Splay Trees with examples. | (10 Marks) | C202.3 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 3. | A) | List the Types of Binary Trees | (2 Marks) | C202.3 | PO1 | 1 |
| B) | Discuss the properties of Red Black tree | (3 Marks) | C202.3 | PO1 | 2 |
| C) | Illustrate AVL Rotations with Examples | (10 Marks) | C202.3 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 4. | A) | Define AVL Tree with example | (2 Marks) | C202.3 | PO1 | 1 |
| B) | Explain about splay tree | (3 Marks) | C202.3 | PO1 | 2 |
| C) | Demonstrate insertion & deletion operation in binary search tree with examples | (10 Marks) | C202.3 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 5. | A) | Define Binary search tree | (2 Marks) | C202.3 | PO1 | 1 |
| B) | Explain the Different rotations in AVL tree | (3 Marks) | C202.3 | PO1 | 1 |
| C) | Demonstrate how to delete an element from AVL Search Trees | (10 Marks) | C202.3 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 6. | A) | Define tree. | (2 Marks) | C202.3 | PO1 | 1 |
| B) | Explain Red-Black tree | (3 Marks) | C202.3 | PO1 | 2 |
| C) | Demonstrate the AVL search tree from following set of elements: 10,12,30,45,47,12,11,5,25,40,13,40,55 | (10 Marks) | C202.3 | PO1,2 3 | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Unit – IV** | **Marks** | **CO** | **PO** | **BTL** |
| 1. | A) | Differentiate linear search and binary search? | (2 Marks) | C202.4 | PO1 | 1 |
| B) | Construct max heap for 80,70,110,100,120,140,130 | (3 Marks) | C202.4 | PO1 | 2 |
| C) | Explain the concept of merge sort in detail and Trace the merge sort algorithm for the following elements:  12,22,54,19,11,84,63,17,15,4,13 | (10 Marks) | C202.4 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 2. | A) | What are the data structures used in BFS and DFS? | (2 Marks) | C202.4 | PO1 | 1 |
| B) | Trace the insertion sort algorithm for the following elements:  3,1,4,7,5,9,2,6,5,10 | (3 Marks) | C202.4 | PO1 | 2 |
| C) | State and explain heap sort. Trace the heap sort algorithm for the following elements:  24,7,17,28,9,1,20,4,26, 21,10 | (10 Marks) | C202.4 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 3. | A) | Apply bubble sort o the following elements:  {21,7,28,9,24} | (2 Marks) | C202.4 | PO1 | 1 |
| B) | Write C program to implement linear search using recursion? | (3 Marks) | C202.4 | PO1 | 2 |
| C) | Explain the working of DFS algorithm. Apply it for the following Graph | (10 Marks) | C202.4 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 4. | A) | Apply selection sort on the following elements:  {21,11,5,78,9,72} | (2 Marks) | C202.4 | PO1 | 1 |
| B) | Write C program to implement binary search using recursion? | (3 Marks) | C202.4 | PO1 | 2 |
| C) | Explain the working of BFS algorithm Illustrate BFS traversal of above graph | (10 Marks) | C202.4 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 5. | A) | Define Heap. Differentiate Max-heap and Min-heap? | (2 Marks) | C202.4 | PO1 | 1 |
| B) | Explain different representations of graphs? | (3 Marks) | C202.4 | PO1 | 1 |
| C) | Write C program for implementing Insertion sort to arrange a list  of integers in ascending order? | (10 Marks) | C202.4 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 6. | A) | Describe divide and conquer methods. | (2 Marks) | C202.4 | PO1 | 1 |
| B) | Write quick sort algorithm. | (3 Marks) | C202.4 | PO1 | 2 |
| C) | Illustrate following data using merge sort.  10,20,6,25,2,36,45,8,9,20 | (10 Marks) | C202.4 | PO1,2 3 | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Unit – V** | **Marks** | **CO** | **PO** | **BTL** |
| 1. | A) | What are the applications of tries? | (2 Marks) | C202.5 | PO1 | 1 |
| B) | Compare the tries with hash table. | (3 Marks) | C202.5 | PO1 | 2 |
| C) | Write and explain Knuth-Morris-Pratt pattern matching algorithm? | (10 Marks) | C202.5 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 2. | A) | What is a tries? Give an example for a tries? | (2 Marks) | C202.5 | PO1 | 1 |
| B) | Explain about failure function in KMP algorithm? | (3 Marks) | C202.5 | PO1 | 2 |
| C) | Write and explain Boyer Moore pattern matching algorithm? | (10 Marks) | C202.5 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 3. | A) | Differentiate standard tries and compressed tries? | (2 Marks) | C202.5 | PO1 | 1 |
| B) | Find the failure function for the pattern”abaca”? | (3 Marks) | C202.5 | PO1 | 2 |
| C) | Write and Explain Brute force pattern matching algorithm and also  analyze its complexity? | (10 Marks) | C202.5 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 4. | A) | What are the advantages and disadvantages of tries? | (2 Marks) | C202.5 | PO1 | 1 |
| B) | Construct a tries for the binary keys 011,111,101,001? | (3 Marks) | C202.5 | PO1 | 2 |
| C) | Explain Brute force algorithm. Using this algorithm, count the total number of comparisons done for the following. Text: babcbabbbcabcabcbbabcabcaacabc Pattern: abcabca. | (10 Marks) | C202.5 | PO1,2 3 | 3 |
|  |  |  |  |  |  |
| 5. | A) | Describe about the pattern matching. List some of the pattern  matching algorithms? | (2 Marks) | C202.5 | PO1 | 1 |
| B) | Consider a Text T=“GCATCGCAGAGAGTATACAGTACG” to match against the pattern P=“AGTATACA” by using the Knuth-Morris-  Pratt pattern algorithm? | (3 Marks) | C202.5 | PO1 | 1 |
| C) | What are tries? Explain in detail about various types of tries with  suitable examples? | (10 Marks) | C202.5 | PO1,2 3 | 3 |
|  |  |  | C202.5 |  |  |

**OBJECTIVE QUESTIONS**

**UNIT – I**

**Multiple Choice Questions**

1. Which of the following points is/are true about Linked List data structure when it is compared with array **[ ]**
2. It is easy to insert and delete elements in Linked List.
3. The size of the array has to be pre-decided, linked lists can change their size any time.
4. Random access is not allowed in a typical implementation of Linked Lists
5. All of the Above
6. In a doubly linked list, the number of pointers affected for an insertion operation at the starting will be **[ ]**
7. 0
8. 1
9. 2
10. 4
11. In a doubly linked list, the number of pointers affected for an insertion operation at the middle will be **[ ]**
12. 0
13. 1
14. 2
15. 4
16. Linked list is considered as an example of \_\_\_\_\_\_\_\_\_\_\_ type of memory allocation. **[ ]**
17. Dynamic
18. Static
19. Compile time
20. Heap
21. Which of these is not an application of a linked list? **[ ]**
22. To implement file systems
23. For separate chaining in hash-tables
24. Expression Conversion
25. Random Access of elements
26. What does ‘stack underflow’ refer to? **[ ]**
27. accessing item from an undefined stack
28. adding items to a full stack
29. removing items from an empty stack
30. None of the above
31. Minimum number of fields in each node of a doubly linked list is \_\_\_\_? **[ ]**
32. 2
33. 3
34. 4
35. None of the above
36. The elements of a linked list are stored **[ ]**
37. In an array
38. In contiguous memory locations
39. Anywhere the computer has space for them
40. All of the above
41. A parentheses checker program would be best implemented using **[ ]**
42. List
43. Queue
44. Stack
45. Any of the above
46. Which of the following data structures works on the principle of First Come First Serve? **[ ]**
47. Array
48. List
49. Stack
50. Queue
51. Which of the following is not a linear data structure? **[ ]**
52. List
53. Stack
54. Queue
55. Binary tree
56. Which of the following data structures permits insertion and deletion operations only on one end of the structure? **[ ]**
57. Linked List
58. Stack
59. Queue
60. Tree
61. Removing element from Queue is known as \_\_\_\_ **[ ]**
62. Push
63. pop
64. Enqueue
65. Dequeue
66. An enqueue operation adds an element \_\_\_\_ **[ ]**
67. At any position in the queue
68. To the front of the queue
69. To the rear of the queue
70. None of above
71. To represent hierarchical relationship between elements, which data structure is suitable **[ ]**
72. List
73. Stack
74. Tree
75. Graph
76. Which type of linked list does not store NULL in the next field? **[ ]**
77. Singly linked list
78. Circular linked list
79. Doubly linked list
80. All of these
81. A linked list is a **[ ]**
82. Random access structure
83. Sequential access structure
84. Both
85. None of these
86. \_\_\_\_\_\_ are appropriate data structures to process a list of employees having a contract for a seniority system for hiring and firing. **[ ]**
87. Queue
88. Stack
89. List
90. Array
91. What is the top element of stack after performing the following operations? insert(5), insert(10),insert(15), insert(111), delete(),insert(11), delete(),delete()**[ ]**
92. 12
93. 11
94. 15
95. 10
96. What is the top element of the queue after performing the following operations? insert(5), insert(10),insert(15), insert(111), delete(),insert(11), delete(),delete()**[ ]**
97. 12
98. 11
99. 15
100. 10

**Fill in the Blanks**

1. The insertion and deletion operations of stack are called \_\_\_\_\_\_\_\_\_\_\_\_\_.
2. \_\_\_\_\_\_\_\_\_\_\_\_ occurs when we try to remove an element from an empty stack.
3. \_\_\_\_\_\_\_\_\_\_data structure is used to remove elements in reverse order.
4. A linear list in which each node has pointers to point to the predecessor and successors nodes is called as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
5. \_\_\_\_\_\_\_\_\_\_\_\_\_ is a data structure used in recursion.
6. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_is the postfix notation of infix 9-((3\*4)+8)/4
7. Data elements in a linked list are known as \_\_\_\_\_\_\_\_\_\_\_\_\_\_.
8. Queue overflow condition: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
9. \_\_\_\_\_\_\_\_\_ stores the address of the first node in the list.
10. In \_\_\_\_\_\_\_\_\_ linked list contains the addressof the last node in the first node.

**UNIT-I KEY**

**MULTIPLE CHOICE**

1) d 2) c 3) d 4) a 5) d 6) c 7) b 8) c

9) c 10) d 11) d 12) b 13) d 14) c

15) c 16) b 17) b 18) a 19) d 20) b

**FILL IN THE BLANKS**

21)Push and Pop 22) Underflow 23) Stack

24)Doubly Linked List 25) Stack 26) 934\*8+4/-27) Nodes 28) rear>MAX-1 29) start

30) Circular Doubly Linked List

**Objective Questions**

**UNIT – II**

**Multiple Choice Questions**

1. Which of the following is not an open addressing technique to resolve collisions? **[ ]**
2. Linear probing
3. Cubic probing
4. Double hashing
5. Quadratic probing
6. In which of the following hash functions, do consecutive keys map to consecutive hash values? **[ ]**
7. Division method
8. Multiplication method
9. Folding method
10. Mid-square method
11. The process of examining memory locations in a hash table is called **[ ]**
12. Hashing
13. Collision
14. Probing
15. Addressing
16. Which open addressing technique is free from clustering problems? **[ ]**
17. Linear probing
18. Quadratic probing
19. Double hashing
20. Rehashing
21. What is a hash table? **[ ]**
22. A structure that maps values to keys
23. A structure that maps keys to values
24. A structure used for storage
25. A structure used to implement stack and queue
26. If several elements are competing for the same bucket in the hash table, what is it called? **[ ]**
27. Diffusion
28. Replication
29. Collision
30. Duplication
31. What is direct addressing? **[ ]**
32. Distinct array position for every possible key
33. Fewer array positions than keys
34. Fewer keys than array positions
35. Same array position for all keys
36. The task of generating alternative indices for a node is called? **[ ]**
37. Collision handling
38. Collision detection
39. Collision recovery
40. Closed hashing
41. Which of the following is not a collision resolution technique? **[ ]**
42. Separate chaining
43. Linear probing
44. Quadratic probing
45. Hashing
46. Which of the following is a disadvantage of using separate chaining using linked lists? **[ ]**
47. It requires many pointers
48. It requires linked lists
49. It uses array
50. It does not resolve collision
51. Which of the following techniques stores data in a separate entity in case of a collision? **[ ]**
52. Open addressing
53. Chaining using doubly linked list
54. Linear probing
55. Double hashing
56. In a hash table of size 10, where is element 7 placed at key\_\_\_\_\_\_\_ with hash function H(k) = (k\*k) mod n? **[ ]**
57. 4
58. 7
59. 49
60. 9
61. What is the advantage of a hash table as a data structure? **[ ]**
62. faster access of data
63. easy to implement
64. very efficient for less number of entries
65. exhibit good locality of reference
66. Load factor of hash table which contains 10 elements when the size is 25? **[ ]**
67. 0.5
68. 0.4
69. 1.0
70. 2.5
71. If the size of a hash table n is 7 then while doing rehashing what is the value of N` ? **[ ]**
72. 49
73. 13
74. 14
75. none of the above
76. Which of the following is the correct function definition for quadratic probing? **[ ]**
77. F(i)=i2
78. F(i)=i
79. F(i)=i+1
80. F(i)=i2+1
81. What is the formula used in Linear probing? **[ ]**
82. Hash key = key mod table size
83. Hash key=(hash(x)+F(i)) mod table size
84. Hash key=(hash(x)+F(i2)) mod table size
85. H(x) = x mod 17
86. What is the hash function used in the division method? **[ ]**
87. h(k) = k/m
88. h(k) = k mod m
89. h(k) = m/k
90. h(k) = m mod k
91. How many steps are involved in creating a hash function using a multiplication method? **[ ]**
92. 1
93. 4
94. 3
95. 2
96. What is the hash function used in the multiplication method? **[ ]**
97. h(k) = floor( m(kA mod 1))
98. h(k) = ceil( m(kA mod 1))
99. h(k) = floor(kA mod m)
100. h(k) = ceil( kA mod m)

**Fill in the Blanks**

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a collection of key-value pairs
2. \_\_\_\_\_\_\_\_\_\_\_ is used to locate the values in the list
3. Every value in a dictionary is assigned with a \_\_\_\_\_\_\_\_\_\_\_\_ key.
4. The main aim of hashing is to perform \_\_\_\_\_\_\_\_\_\_\_\_\_\_ operation in O(1) time.
5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a mathematical formula which is applied to key to produce an index in hash table
6. \_\_\_\_\_\_\_\_\_\_\_\_ occurs when a hash function maps two different keys to the same location.
7. The presence of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the hash table indicates that the location has no data at present.
8. The process of examining memory locations in the hash table is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
9. Probe sequence ranges from \_\_\_\_\_\_\_\_\_ to \_\_\_\_\_\_\_\_\_.
10. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_ are two collision resolution techniques.

**UNIT-II KEY**

**MULTIPLE CHOICE**

1) b 2) a 3) c 4) c 5) b 6) c 7) a

8) a 9) d 10) a 11) b 12) d 13) a 14) b

15) b 16) a 17) b 18) b 19) d 20) a

**FILL IN THE BLANKS**

21) Dictionary 22) key 23) Unique

24) Search 25) Hash Function 26) Collision

27) Sentinel value(-1) 28) Probing 29) 0 to m-1

30) Open addressing and chaining

**Objective Questions**

**UNIT – III**

**Multiple Choice Questions**

1. In a Tree Degree of a leaf node is\_\_\_\_\_ **[ ]**
2. 0
3. 1
4. 2
5. 3
6. The binary tree of height h has at least **h+1** nodes and at most \_\_\_\_\_ nodes **[ ]**
7. 2h
8. 2h
9. 2h+1
10. 2h+1-1
11. Maximum no. of nodes at the nth level of a binary tree can be **[ ]**
12. 2n
13. 2n
14. 2n+1
15. 2n-1
16. In a Tree Representation with array if **root node index is 0** then in that tree for element at position **i** left child stored at position **[ ]**
17. 2i
18. 2i+1
19. i/2
20. 2n-1
21. A binary search tree is also known as **[ ]**
22. Complete Binary Tree
23. Full Binary Tree
24. Ordered Binary Tree
25. None of the above
26. AVL & Red-Black Treesare also known as **[ ]**
27. Complete BST
28. Full BST
29. Ordered BST
30. None of the above
31. At a Particular node of the AVL Tree in balance is happened becauseof left sub tree then Probably the balance factor is **[ ]**
32. -2
33. -1
34. 1
35. 2
36. Which rotation is done when the new node is inserted in the right sub-tree of the right sub-tree of the critical node? **[ ]**
37. LL
38. LR
39. RR
40. RL
41. Which rotation is done when the new node is inserted in the right sub-tree of the left sub-tree of the critical node? **[ ]**
42. LL
43. LR
44. RR
45. RL
46. When a Node N is accessed it is splayed to make it the **[ ]**
47. Root Node
48. Sibling Node
49. Child Node
50. Parent Node

**Fill in the Blanks**

1. The maximum number of nodes at the nth level of a binary tree is \_\_\_\_\_\_\_\_\_\_.
2. Nodes at the same level that share the same parent are called as \_\_\_\_\_\_\_\_\_\_\_\_\_\_.
3. Leaf nodes are also called as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
4. A binary tree of n nodes has exactly \_\_\_\_\_\_\_\_\_\_\_ edges.
5. The average running time of a search operation in binary search tree is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
6. An AVL tree is a self-balancing tree which is also known as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
7. Every node of an AVL tree has a balance factor of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
8. A Red-Black tree is a self-balancing tree which is also known as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
9. In a red-black tree, the colour of a root node is \_\_\_\_\_\_\_\_\_\_ and the colour of a leaf node is \_\_\_\_\_\_\_\_\_\_\_\_.
10. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a self-balancing search tree with an additional property that recently accessed elements can be re-accessed fast.

**UNIT-III KEY**

**MULTIPLE CHOICE**

1) a 2) d 3) b 4) b 5) c

6) c 7) d 8) C 9) b 10) a**FILL IN THE BLANKS**

11) 2n  12) Siblings 13) External nodes

14) n-1 15) O(log n) 16) Height-balanced tree

17) -1 , 0 , -1 18) Symmetric binary B-tree 19) Black, Black 20) Splay Tree

**UNIT – IV**

**Multiple Choice Questions**

1. The number of edges that originate at an node are called\_\_\_\_\_\_\_\_ of that node**[ ]**
2. In-degree
3. Out-degree
4. Degree
5. All of the Above
6. \_\_\_\_\_\_\_\_\_\_\_\_ sorting algorithm is of divide and conquer type? **[ ]**
7. Heap sort
8. Insertion sort
9. Merge sort
10. Selection sort
11. In a Directed Graph Number of edges equal to \_\_\_\_\_\_\_\_\_\_\_of all nodes **[ ]**
12. sum of In-Degree
13. sum of out-Degree
14. sum of Both In-Degree and Out-Degree
15. sum of In-Degree or Out-Degree
16. In a Undirected Graph Number of edges equal to \_\_\_\_\_\_\_\_\_\_\_ of all nodes **[ ]**
17. sum of Degree
18. half the sum of Degree
19. Both a and b
20. None of the above
21. Graphs can be represented with **[ ]**
22. Array
23. List
24. Both a and b
25. None of the above
26. Best Example for External Sorting is\_\_\_\_\_\_\_\_\_ **[ ]**
27. Bubble Sort
28. Insertion Sort
29. Heap Sort
30. Merge Sort
31. In a Max Heap Tree \_\_\_\_\_\_\_\_\_\_ **[ ]**
32. root node value is greater than chelids
33. root node is value less than chelids
34. Both A and B
35. None of the above
36. \_\_\_\_\_\_\_\_\_number of edges present in a complete graph having n vertices. **[ ]**
37. n
38. n-1
39. (n\*(n+1))/2
40. (n\*(n-1))/2
41. For a given graph G having v vertices and e edges which is connected and has no cycles, which of the following statements is true? **[ ]**
42. v=e
43. v = e+1
44. v + 1 = e
45. v = e-1
46. A graph with all vertices having equal degree is known as a \_\_\_\_\_\_\_\_\_\_ **[ ]**
47. Regular Graph
48. Multi Graph
49. Simple Graph
50. Complete Graph

**Fill in the Blanks**

1. \_\_\_\_\_\_\_\_\_\_\_\_ is a process by which two ordered list of elements are combined into single ordered list.
2. Time complexity of binary search is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. \_\_\_\_\_\_\_\_\_ is the time complexity of heap sort.
4. In-degree of a node ‘u’ is the number of edges that \_\_\_\_\_\_\_\_\_\_\_ at ‘u’.
5. Adjacency matrix is also known as \_\_\_\_\_\_\_\_\_\_\_\_\_\_.
6. Depth First Search in graph is like \_\_\_\_\_\_\_\_\_ traversal of a tree.
7. Breadth First Search in graph is like \_\_\_\_\_\_\_\_\_ traversal of a tree.
8. Depth First Search uses \_\_\_\_\_\_\_\_\_\_\_ data structure.
9. Breadth First Search uses \_\_\_\_\_\_\_\_\_\_\_ data structure.
10. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is an extension of linked representation of graph.

**UNIT-IV KEY**

**MULTIPLE CHOICE**

1) b 2) c 3) d 4) b 5) c

6) d 7) a 8) d 9) b 10) a

**FILL IN THE BLANKS**

11) Merging 12) O(log n) 13) O(n log n)

14) ends 15) Bit/Boolean matrix 16) Pre-order

17) Level-order 18) Stack 19) Queue

20) Adjacency Multi List

**UNIT – V**

**Multiple Choice Questions**

1. Pattern matching refers to string [ ]

a)searching b)matching

c)Both a & b d)None of the above

1. What is the worst case time complexity of KMP algorithm for pattern searching (m = length of text, n = length of pattern)? [ ]  
   a) O(n) b) O(n\*m) c) O(m) d) O(log n)
2. What is the worst case time complexity of Brute Force algorithm for pattern searching (m = length of text, n = length of pattern)? [ ]  
   a) O(n) b) O(n\*m) c) O(m) d) O(log n)
3. What is the worst case time complexity of Boyer Moore algorithm for pattern searching (m = length of text, n = length of pattern)? [ ]  
   a) O(n) b) O(n\*m) c) O(m) d) O(log n)

1. Which of the following is used in KMP algorithm [ ]

a) Prefix array b) Suffix array

c) Both a & b d) None of these

1. The concept of prefix and suffix is used in which of the following algorithms? [ ]

a) KMP b) Brute-Force

c) Boyer Moore d) None of these

1. Trie is also known as \_\_\_\_\_\_\_\_\_ [ ]  
   a) Digital Tree b) Treap  
   c) Binomial Tree d) 2-3 Tree
2. A program to search a contact from phone directory can be implemented efficiently using \_\_\_\_\_\_ [ ]  
   a) a BST  
   b) a trie  
   c) a balanced BST  
   d) a binary tree
3. What can be the maximum depth of the trie with n strings and m as themaximum sting the length? [ ]  
   a) log2n  
   b) log2m  
   c) n  
   d) m
4. Which of the following is true about the trie? [ ]  
   a) root is letter a  
   b) path from root to the leat yields the string  
   c) children of nodes are randomly ordered  
   d) each node stores the associated keys

**Fill in the Blanks**

1. A \_\_\_\_\_\_\_\_\_\_\_ is a tree-based date structure for storing strings in order to make pattern matching faster.
2. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_ is like a standard trie but makes sure that each trie had a degree of at least 2.
3. \_\_\_\_\_\_\_\_\_\_\_\_ array is used in KMP algorithm.
4. \_\_\_\_\_\_\_\_\_\_\_traversal over trie gives the lexicographical sorting of the set of the strings.
5. \_\_\_\_\_\_\_\_\_is the efficient data structure for searching words in dictionaries.
6. Auto complete and spell checkers can be implemented efficiently using the trie. \_\_\_\_\_\_\_\_\_\_ (True/Flase)
7. Best case time complexity of brute force algorithm if pattern found is \_\_\_\_\_\_\_\_ (Length of text is N and Length of Pattern is M).
8. Best case time complexity of brute force algorithm if pattern not found is \_\_\_\_\_\_\_\_ (Length of text is N and Length of Pattern is M).
9. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a special type of trie is used for fast searching of the full texts.
10. Trie requires \_\_\_\_\_\_\_\_\_\_\_\_storage space than hashing

**UNIT-V KEY**

**MULTIPLE CHOICE**

1) c 2 ) a 3) b 4) b 5) a

6) a 7 ) a 8) b 9) d 10) b

**FILL IN THE BLANKS**

11)Trie 12)Compressed Trie 13)Prefix 14)In-order

15)Trie 16) True 17)O(M) 18)O(N)

19)Suffix Trie 20)less