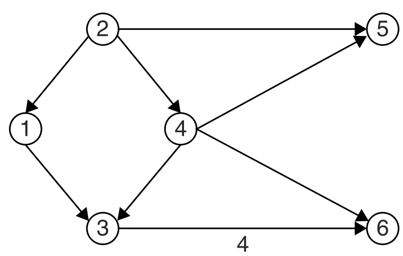
Chapter 3

Greedy Method

One-mark Questions

1. Consider the DAG with
$V = \{1, 2, 3, 4, 5, 6\}$, shown below.



Which of the following is NOT a topological ordering? **[2009]**
(a) 1 2 3 4 5 6
(b) 1 3 2 4 5 6
(c) 1 3 2 4 6 5
(d) 3 2 4 1 6 5

Solution: (d)
1 should come before 3, 2 and 4 as shown by the edges in the DAG.
Hence, the correct option is (d).

2. Consider a weighted complete graph G on the vertex set $\{v_1, v_2, v_n\}$ such that the weight of, the edge (v_i, v_j) is $2\ |i - j|$. The weight of a minimum spanning tree of G is: **[2006]**
(a) $n - 1$
(b) $2n - 2$
(c) $\frac{n}{2}$
(d) n^2

Solution: (b)
When v_i is connected to v_{i+1}, the edge cost is 2. there will be $(n - 1)$ such edges with a cost of '2'.
Therefore total cost $= 2(n - 1) = 2n - 2$
(or)

$$\sum_{i=1}^{n} 2\left|\vartheta_i - \vartheta_i - 1\right| = 2\sum_{i=1}^{n}\left|1\right| = 2\left|n-1\right| = 2n-2 \qquad (1)$$

Hence, the correct option is (b).

3. To implement Dijkstra's shortest path algorithm on weighted graphs so that it runs in linear time, the data structure to be used is: **[2006]**
(a) Queue
(b) Stack
(c) Heap
(d) B-Tree

Solution: (c)
Edges and vertices of the graph can be represented in the form of a Heap data structure to have a linear time complexity algorithm.
Hence, the correct option is (c).

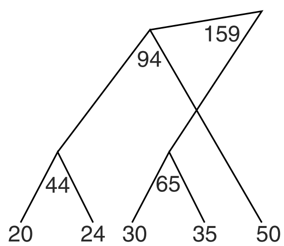
Two-marks Questions

1. Suppose P, Q, R, S, T are sorted sequences having lengths 20, 24, 30, 35, 50 respectively. They are to be merged into a single sequence by merging together two sequences at a time. The number of comparisons that will be needed in the worst case by the optima algorithm for doing this is___ **[2014]**

Solution: 358

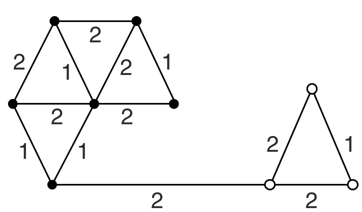
If one sorted list contain 'm' elements and another sorted list contain 'n' elements then in the worst case we can merge them into a single sorted list with $(m + n - 1)$ comparison.

Hence, given files are P, Q, R, S, T

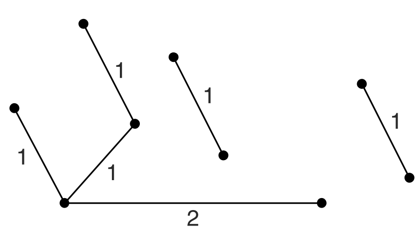


Total number of comparisons = 158 + 93 + 64 + 43 = 358.

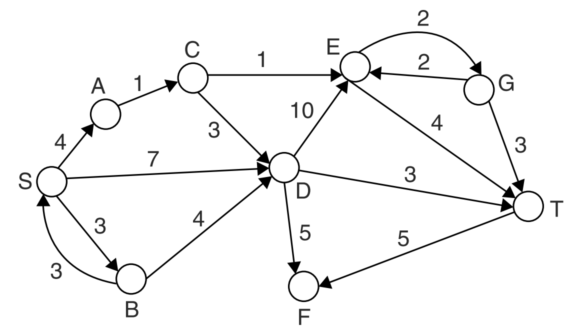
2. The number of distinct minimum spanning trees for the weighted graph below is_____ **[2014]**



Solution: 6



3. Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T. Which one will be reported by Dijkstra's shortest path algorithm? Assume that, in any iteration, the shortest path 0 a vertex v is updated only when a strictly shorter path to v is discovered. **[2012]**



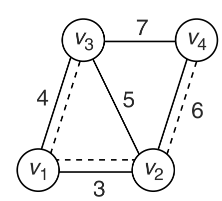
(a) SDT (b) SBD
(c) SACDT (d) SACET

Solution: (d)

In Dijkstra's Algorithm, we always consider the vertex which are go from the source with last cost, and update cost lable of the vertex, if the present cost is minimum than the previous cost. Greedy based Dijkstra's Algorithm can be used here.

Hence, the correct option is (d).

4. An undirected graph $G(V, E)$ contains $n(n > 2)$-named $v_1, v_2, \ldots v_n$. Two nodes vi, vj are connected if and only if $0 < |i - j| \leq 2$. Each edge (v_i, v_j) assigned a weight $i + j$. A sample graph with is shown below. **[2011]**



What will be the cost of the minimum spanning tree (MST) of such a graph with n nodes?

(a) $\frac{1}{12}(11n^2 - 5n)$ (b) $n^2 - n + 1$
(c) $6n$ (d) $2n + 1$

Solution: (b)

Cost of MST with '4' vertices is 3 + 4 + 6

Cost of MST with '5' vertices is 3 + 4 + 6 + 8

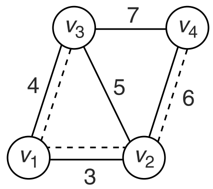
In general cost for 'n' vertices we have

$= 3 + 4 + 6 + 8 + \ldots + 2n - 2$

$= n^2 - n + 1$

Hence, the correct option is (b).

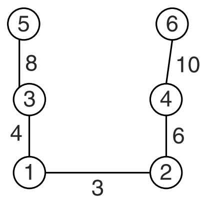
5. An undirected graph $G(V, E)$ contains $n(n > 2)$-named $v_1, v_2, \ldots v_n$. Two nodes v_i, v_j are connected if and only if $0 < |i - j| \le 2$. Each edge (v_i, v_j) assigned a weight $i + j$. A sample graph with is shown below. **[2011]**



The length of the path from v_5 to v_6 in the MST of previous question with $n = 10$ is

(a) 11 (b) 25 (c) 31 (d) 41

Solution: (c)



Hence, the correct option is (c).

6. Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry $W(i, j)$ in the matrix W below is the weight of the edge $\{i, j\}$ **[2010]**

$$W = \begin{bmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{bmatrix}$$

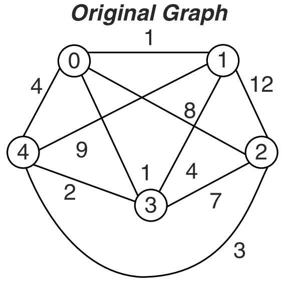
What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T?

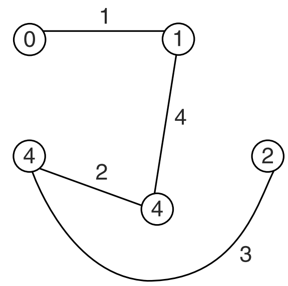
(a) 7 (b) 8 (c) 9 (d) 10

Solution: (d)

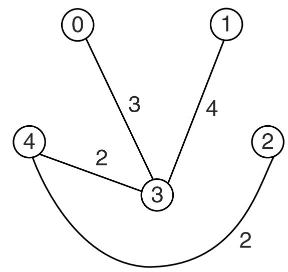
Two possible Spanning Trees wherein vertex 'o' is at the leaf level is as given below:

Original Graph





Tree - 1



Tree - 2

Hence, the correct option is (d).

7. Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry $W(i, j)$ in the matrix W below is the weight of the edge $\{i, j\}$ **[2010]**

$$W = \begin{bmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{bmatrix}$$

What is the minimum possible weight of a path P from vertex 1 to vertex 2 in this graph such that P contains at most 3 edges?

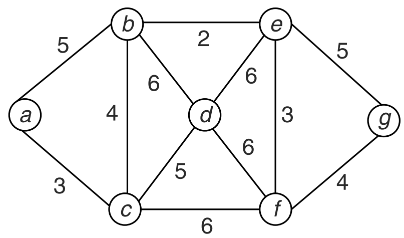
(a) 7 (b) 8 (c) 9 (d) 10

Solution: (c)

With 4 edges it is possible to have a cost of (7) for the path between 1 and 2. But this path 'p' will have 4 edges i.e. (1-0-3-4-2). This is obtained using Dijkstra's Single Source shortest path spanning tree. However with almost 3 edges, the minimum possible cost path from 1 to 2 is not less than 9.

Hence, the correct option is (c).

8. Consider the following graph:



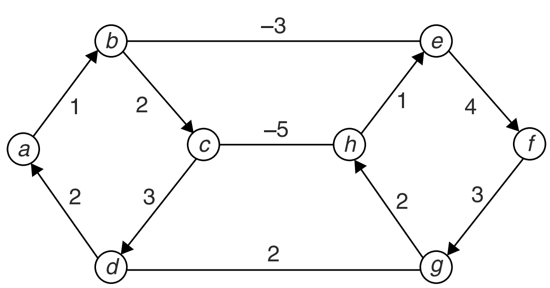
Which one of the following is NOT the sequence of edges added to the minimum spanning tree using Kruskal's algorithm? **[2009]**

(a) (b, e) (e, f) (a, c) (b, c) (f, g) (c, d)
(b) (b, e) (e, f) (a, c) (f, g) (b, c) (c, d)
(c) (b, e) (a, c) (e, f) (b, c) (f, g) (c, d)
(d) (b, e) (e, f) (b, c) (a, c) (f, g) (c, d)

Solution: (d)

After adding (b, e) one can add either (e, f) or (a, c) Kruskal's algorithm uses greedy strategy (picks minimum weight edge). Weight of edge (a, c) is less than (b, c). So it cannot come after (b, c).
Hence, the correct option is (d).

9.



Dijkstra's single source shortest path algorithm when run from vertex a in the above graph, computes the correct shortest path distance to **[2008]**

(a) only vertex a
(b) only vertices a, e, f, g, h
(c) only vertices a, b, c, d
(d) all the vertices

Solution: (c)

Dijkstra's Algorithm can not apply for negative weight edges.
Hence, the correct option is (c).

10. Let w be the minimum weight among all edge weights in an undirected connected graph. Let e be a specific edge of weight w. Which of the following is FALSE? **[2007]**

(a) There is a minimum spanning tree containing e.
(b) If e is not in a minimum spanning tree T, then in the cycle formed by Adding e to T, all edges have the same weight.
(c) Every minimum spanning tree has an edge of weight w.
(d) e is present in every minimum spanning tree.

Solution: (d)

If there are multiple edges in the graph with the minimum weight 'w', then it is not necessary that the specific edge with cost 'w' must be present in all spanning trees. There may be many edges of weight w in the graph and e.
Hence, the correct option is (d).

11. Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32}$, respectively.

Which of the following is the Huffman code for the letter a, b, c, d, e, f? **[2007]**

(a) 0, 10, 110, 1110, 11110, 11111
(b) 11, 10, 011, 010, 001, 000
(c) 11, 10, 01, 001, 0001, 0000
(d) 110, 100, 010, 000, 001, 111

Solution: (a)

The Huffman Encode tree can be obtained by applying optimal merge pattern algorithm. Assign '0' to the left branch and '1' to the right branch in the encode tree. Collect the stream of binary bits to get the codes of the message.
Hence, the correct option is (a).

12. Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32}$, respectively. What is the average length of the correct answer to Q.11? **[2007]**

(a) 3 (b) 2.1875
(c) 2.25 (d) 1.9375

Solution: (d)

The average no. of bits/message is obtained by using the formula

$$\sum_{i=1}^{n} d_i {}^* q_i$$

d_i = distance from root to message i
q_i = probability of message i
Hence, the correct option is (d).

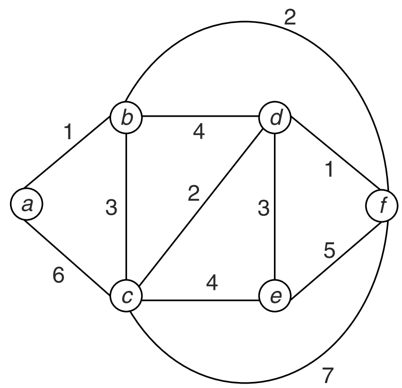
13. In an unweighted, undirected connected graph, the shortest path from a node S to every other node is computed most efficiently, in terms of time complexity by **[2007]**

(a) Dijkstra's algorithm starting from S.
(b) Warshall's algorithm
(c) Performing a DFS starting from S.
(d) Performing a BFS starting from S.

Solution: (d)

It is one of the application of BFS.
Hence, the correct option is (d).

14. Consider the following graph:



Which one of the following cannot be the sequence of edges added, in that order, to a minimum spanning tree using Kruskal's algorithm? **[2006]**

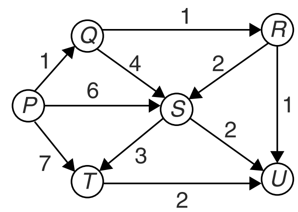
(a) (a–b), (d–f), (b–f), (d–c), (d–e)
(b) (a–b), (d–f), (d–c), (b–f), (d–e)
(c) (d–f), (a–b), (d–c), (b–f), (d–e)
(d) (d–f), (a–b), (b–f), (d–e), (d–c)

Solution: (d)

(a) $(a^1 - b)$, $(d^1 - f)$, $(b^2 - f)$, $(d^2 - c)$, $(d^3 - e)$ - correct
(b) $(a^1 - b)$, $(d^1 - f)$, $(d^2 - f)$, $(b^2 - f)$, $(d^3 - e)$ - correct
(c) $(d^1 - f)$, $(a^1 - b)$, $(d^2 - c)$, $(b^2 - f)$, $(d^3 - e)$ - correct
(d) $(d^1 - l)$, $(a^1 - b)$,$(b - f^2)$,$(d - e^3$ $(d^2 - c)$ - incorrect

Hence, the correct option is (c).

15. Suppose we run Dijkstra's single source shortest-path algorithm on the following edge-weighted directed graph with vertex P as the source.



In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized? **[2004]**

(a) P, Q, R, S, I, U
(b) P, Q, R. U, S, T
(c) P, Q, R, U, T, S
(d) P, Q, T, R, U, S

Solution: (b)

(a) $\underset{1}{P \; Q} \; \overset{2}{Q \; R} \; \underset{4}{R \; S} \; \overset{7}{S \; T} \; \underset{3}{T \; U}$ - Invalid (not in increasing order)

(b) $\underset{1}{P \; Q} \; \overset{2}{Q \; R} \; \underset{3}{R \; U} \; \overset{4}{U \; S} \; \underset{4}{S \; T}$ - Valid

(c) $\underset{1}{P \; Q} \; \overset{2}{Q \; R} \; \underset{3}{R \; U} \; \overset{7}{U \; T} \; \underset{4}{T \; S}$ - Invalid

(d) $\underset{1}{P \; Q} \; \overset{7}{Q \; T} \; \underset{2}{T \; R} \; \overset{3}{R \; U} \; \underset{4}{U \; S}$ - Invalid

Hence, the correct option is (b).

16. Let $G = (V, E)$ be an undirected graph with a subgraph $G_I = (V_I, E_I)$. Weights are assigned to edges of G as follows.

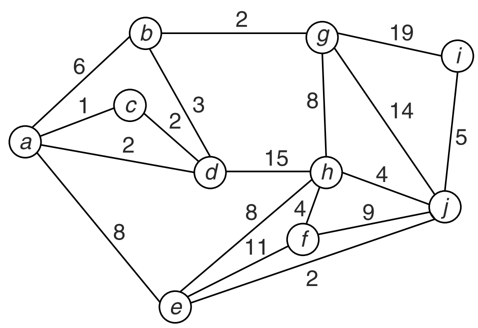
$$w(e) = \begin{cases} 0 \; if \; e \in E_1 \\ 1 \; otherwise \end{cases}$$

A single-source shortest path algorithm is executed on the weighted graph (V, E, w) with an arbitrary vertex v_i of V_1 as the source. Which of the following can always be inferred from the path costs computed? **[2003]**

(a) The number of edges in the shortest paths from v_i to all vertices of G
(b) G_i is connected
(c) V_i forms a clique in G
(d) G_i is a tree

Solution: (a)
No. of edges in the shortest paths can be determined as a result of Dijkstra's single source shortest paths algorithm.
Hence, the correct option is (a).

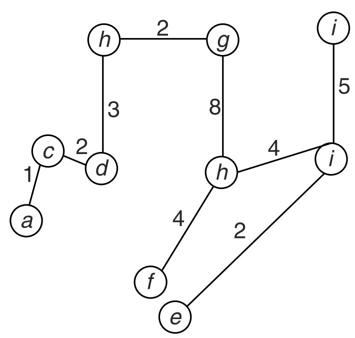
17. What is the weight of a minimum spanning tree of the following graph?



[2003]

(a) 29 (b) 31
(c) 38 (d) 41

Solution: (b)



Weight

Hence, the correct option is (b).

18. Let $G = (V, E)$ be a directed graph with n vertices. A path from v_i to v_j in G is a sequence of vertices $(v_i, v_{i+1}, \ldots, v_j)$ such that $(v_k, v_{k+1}) \in E$ for all k in i through $j - 1$. A simple path is a path in which no vertex appears more than once.
Let A be an $n \times n$ array initialized as follows.

$$A[j,k] = \begin{cases} 1 \; if \; (j,k) \in E \\ 1 \text{ otherwise} \end{cases}$$

Consider the following algorithm.

```
for i = 1 to n
for j = 1 to n
for k = 1 to n
A[j, k] - max(A[j, k], A[j, i] + A[i,
 k]);
```

Which of the following statements is necessarily true for all j and k after termination of the above algorithm? **[2003]**

(a) $A[j, k] < n$
(b) If $A[j, j] > n - 1$, then G has a Hamiltonian cycle
(c) If there exists a path from j to k, $A[j, k]$ contains the longest path length from j to k
(d) If there exists a path from j to k, every simple path from j to k contains at most $A[j, k\}$ edges

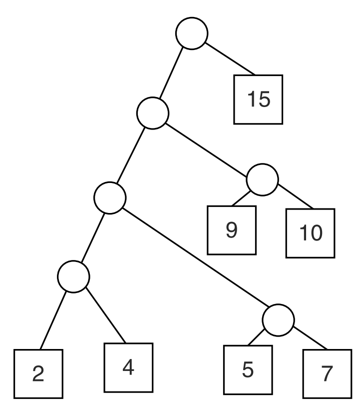
Solution: (c)
It is just opposite of all pairs shortest path, where max replaces min; therefore it finds longest path cost from j to k;
Hence, the correct option is (c).

19. Let G be an undirected connected graph with distinct edge weight. Let max e be the edge with maximum weight and min e the edge with minimum weight. Which of the following statements is false? **[2000]**

(a) Every minimum spanning tree of G must contain mine
(b) If max e is in a minimum spanning tree, then its removal must disconnect G
(c) No minimum spanning tree contains max e
(d) G has a unique minimum spanning tree

Solution: (d)
Hence, the correct option is (d).

20. The weighted external path length of the binary tree in figure is **[1991]**



Solution: The formula for weighted external path length

$= \sum_{i=1}^{n} d_i{}^{*} q_i ; d_i$ = dist. from root to

F_i q_i = length of file F_i.

$= 4 \times 2 + 4 \times 4 + 4 \times 5 + 4 \times 7 + 3 \times 9 + 3 \times 10 + 1 \times 15$

$= 8 + 16 + 20 + 28 + 27 + 30 + 15 = 144$

21. Kmskal's algorithm for finding a minimum spanning tree of a weighted graph G with n vertices and m edges has the time complexity of: **[1991]**

(a) $O(n^2)$ (b) $O(mn)$

(c) $O(m + n)$ (d) $O(m \log m)$

Solution: (d)

Kruskal's Algorithm uses min Heap to keep the list of edges 9(m log m). The sorting of the edges can be done in $O(m\log n)$ which is $O(m\log n)$ for any graph because for each edge (u, v) we check whether u and v are in the same tree, this is done with two calls to find which is $O(\log n)$, and we union the two if necessary which is $O(1)$. Therefore the loop is $O(m\log n)$. Hence the total time complexity is $O(m\log n)$.

Hence, the correct option is (d).

22. Complexity of Kruskal's algorithm for finding the minimum spanning tree of an undirected graph containing n vertices and m edges if the edges are sorted is________

Solution: θ(m log m); 'm' edges are maintained in the form of min-Heap Heap operation requires log m time for deletion. Hence 9(m log m)

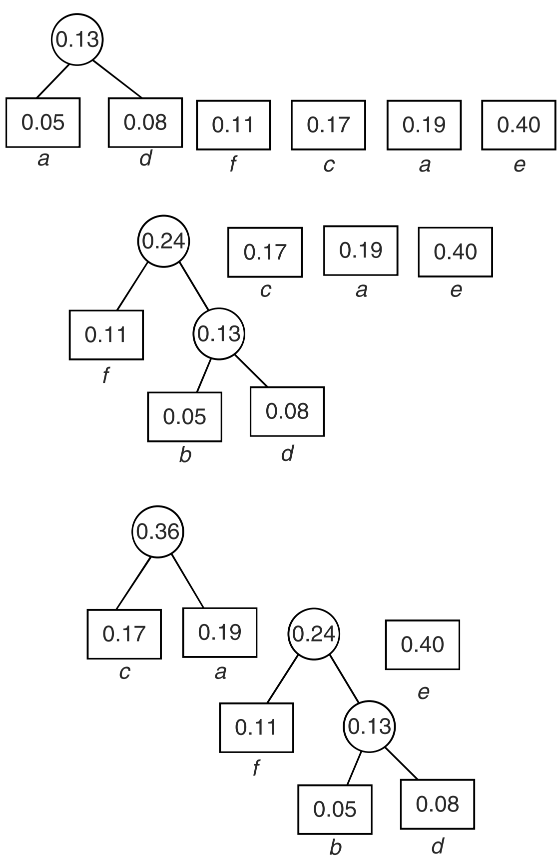
Five-marks Questions

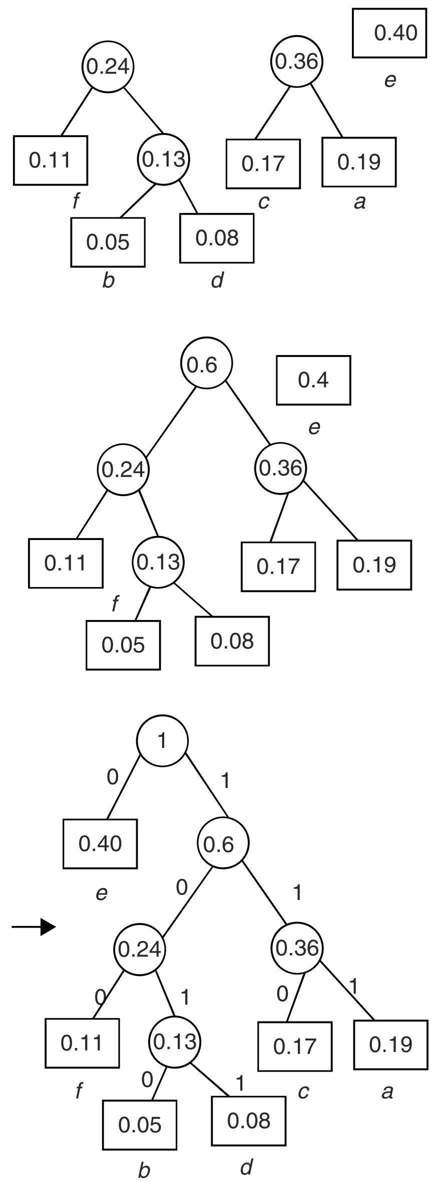
1. A Language uses an alphabet of six letters, $\{a, b, c, d, e, f\}$. The relative frequency of use of each letter of the alphabet in the language is m given below.

Letter	Relative frequency of use
A	0.19
B	0.05
C	0.17
D	0.08
E	0.40
F	0.11

Design a prefix binary code for the language which would minimize the average length of the encoded words of the language. **[1989]**

Solution:





Prefix codes:
$a \to 111$; $b \to 1010$;
$c \to 110$; $d \to 1011$; $e \to 0$

2. An independent set in a graph is a subset of vertices such that no two vertices in the subset are connected by an edge. An incomplete scheme for a greedy algorithm to find a maximum independent set in a tree is given below:

```
V: Set of all vertices in the tree;
   I:= Φ;
While V ≠ O do
begin
select a vertex u; e V such that
V:=V − {u};
if u is such that
then I: = IU {u}
end;
output(I);
```

(a) Complete the algorithm by specifying the property of vertex u in each case

(b) What is the time complexity of the algorithm. **[1994]**

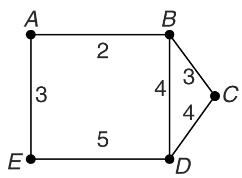
Solution:

(a) (1) $\in$ I

(2) u is not adjacent to any vertex of I

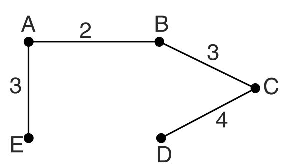
(b) time complexity is $O(n^2)$

3. How many minimum spanning trees does the following graph have? Draw them. (Weights are assigned to the edge). **[1995]**



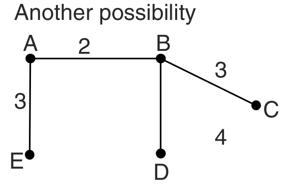
Solution:

By using Kruskal's Algorithm



Cost = 12

Another possibility



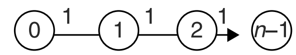
4. A complete, undirected, weighted graph G is given on the vertex $\{0, 1 \ldots, n-1\}$ for any fixed 'n'. Draw the minimum spanning tree of G if

(a) the weight of the edge (u, v) is $|u - v|$

(b) the weight of the edge (u, v) is $u + v$ **[1996]**

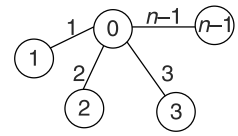
Solution:

(a)



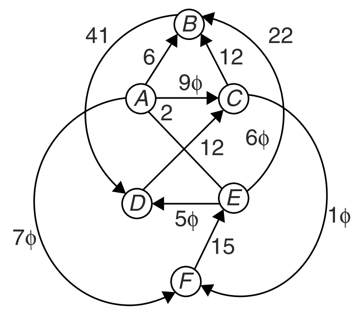
It is a chain of $(n-1)$ edges in an n - vertex graph skewed tree.

(b)



It is like a star graph.

5. Let G be the directed, weighted graph shown below in Figure



We are interested in the shortest paths from A.

(a) Output the sequence of vertices identified by the Dijkstra's algorithm for single source shortest path when the algorithm is started at node A to E.

(b) Write down sequence of vertices in shortest path from A to E.

(c) What is the cost of the shortest path from A to E? **[1996]**

Solution:

(a) Sequence of vertices identified by the Dijkstra's Single Source shortest path algorithm when started from vertex 'A' is A, F, B, E, C

(b) Shortest path from A to E is A, B, D, C, E

(c) Cost = 69 (6 + 41 + 12 + 10)

6. Consider a graph whose vertices are points in the plane with integer co-ordinates (x, y) such that $1 \leq x \leq n$ and $1 \leq y \leq n$, where $n \geq 2$ is an integer. Two vertices (x_1, y_1) and (x_2, y_2) are adjacent if $|x_1 - x_2| \leq 1$ and $|y_1 - y_2| \leq 1$. The weight of an edge $\{(x_1, y_1).$ $(x_2, y_2)\}$ is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

(a) What is the weight of a minimum weight-spanning tree in this graph? Write only the answer without any explanations.

(b) What is the weight of a maximum weight-spanning tree in this graph? Write only the answer without any explanations? **[1997]**

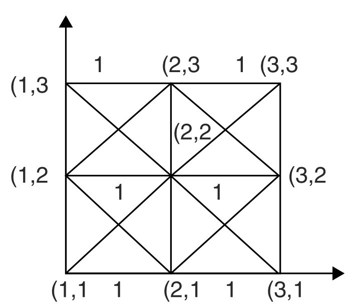
a) A, E, B, D, C, F

b) A to E $\Rightarrow$ A $\rightarrow$ E

c) cost (A, E) = 2

Initial	S	Value X schedule	DISTANCE					
			A	B	C	D	E	F
1	A	{E}		6	9		2	7
	6	6	5	2	7			
2	A,E	{B}		6	6	4	2	7
3	A,B,E	{D}		6	5	4	2	7
4	A, B, E, D	{C}		6	5	4	2	6
5	A, B, C, D, E	F		6	5	4	2	6

Let us take $n = 3$, then the graph can be shown as



all edge costs are unit cost form the given condition (a) and (b) since a spanning tree with n^2-vertices will have $(n^2 - 1)$ edges.

Therefore cost of both minimum and maximum sparing tree is $(n - 1)$ and $\sqrt{2}\left(n^2 - 2\right) + 1$ respectively.

7. Consider a weighted undirected graph with vertex set

$V = \{n_1, n_2, n_3, n_4, n_5$ And Edge set $E = \{(n_1, n_2, 2), (n_1, n_3, 8), (n_1, n_6, 3), (n_2, n_4, 4), (n_2, n_5, 12), (n_3, n_4, 7), (n_4, n_5, 9), (n_4, n_6, 4)\}$. The third value in each tuple represents the weight of the edge specified in the tuple.

(a) List the edges of a minimum spanning tree of the graph.

(b) How many distinct minimum spanning trees does this graph have?

(c) Is the minimum among the edge weights of a minimum spanning tree unique overall possible minimum spanning trees of a graph?

(d) Is the maximum among the edge weights of a minimum spanning tree unique over all possible minimum spanning trees of a graph? **[2001]**

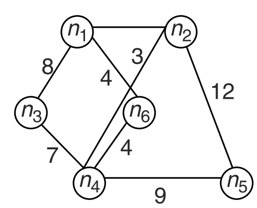
Solution:

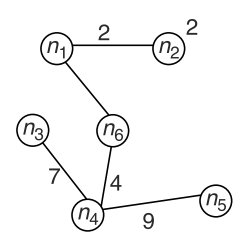
(a) $[(n_1, n_2), (n_1, n_6), (n_4, n_6), (n_3, n_4), (n_4, n_5)]$

(b) It has two distinct Spanning Tree

(c) Yes it is unique;

The graph and its Spanning Tree





Chapter 4

Dynamic Programming

One-mark Questions

1. An algorithm to find the length of the longest monotonically increasing sequence of numbers in an array

$A[0: n-1]$ is given below.

Let L_i denote the length of the longest monotonically increasing sequence starting at index i in the array

Initialise $L_{n-1} = 1$

For all i such that $0 \le i \le n-2$

$$L_i = \begin{cases} 1 + L_{i+1} & \text{if } A[i] < A[i+1] \\ 1 & \text{otherwise} \end{cases}$$

Finally the length of the longest monotonically increasing sequence is

$$\text{Max}(L_0, L_1, \ldots L_{n-1})$$

Which of the following statements is TRUE? **[2011]**

(a) The algorithm uses dynamic programming paradigm

(b) The algorithm has a linear complexity and uses branch and bound paradigm

(c) The algorithm has a non-linear polynomial complexity and uses branch and bound paradigm

(d) The algorithm uses divide and conquer paradigm.

Solution: (a)

It is based on the principle of optimality.

Hence, the correct option is (a).

2. The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of **[2004]**

(a) n (b) n^2

(c) $n \log n$ (d) $n \log^2 n$

Solution: (c)

The number of comparisons that a comparison sort algorithm requires in proportion to $n \log n$, where n is the number of elements to sort.

Hence, the correct option is (c).

3. Which one of the following algorithm design techniques is used in finding all pairs of shortest distances in a graph? **[1998]**

(a) Dynamic programming

(b) Backtracking

(c) Greedy

(d) Divide and Conquer

Solution: (a)

Hence, the correct option is (a).

Two-mark Questions

1. A sub-sequence of a given sequence is just the given sequence with some elements (possibly none or all) left out. We are given two sequences $X[m]$ and $Y[n]$ of lengths m and n, respectively, with indexes of X and Y starting from 0 **[2009]**

We wish to find the length of the longest common sub-sequence (LCS) of $X[m]$ and $Y[n]$ as $l(m, n)$, where an incomplete recursive definition for the function $l(i, j)$ to compute the length of the LCS of $X[m]$ and $Y[n]$ is given below:

$l(i, j) = 0$, if either $i = 0$ or $j = 0$

= exprl, if i, j > 0 and

$x[i - ---l] = y[j - l]$
$=$ expr2, if $i, j > 0$ and
$x[i - 1] \neq y[j - 1]$
Which one of the following options is correct?
(a) expr1 $\equiv l(i - lj) + l$
(b) expr2 $\equiv l(i, j - l)$
(c) expr2 $\equiv \max(l(i - 1, j), l(i, j - l))$
(d) expr2 $\equiv \max(l(i - l, j - 1), l(i, j))$
Using LCS algorithm, in Dynamic programming we can write
expr1 $= l + l(i - l\, j - 1)$; and
expr2 $= \max(l(i - 1, j), l(i, j - 1))$

Solution: (b)
Hence, the correct option is (b).

2. The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_1, a_2, a_3, \ldots, a_n\}$, and positive integer W, is there a subset of S whose elements sum to W? A dynamic programme for solving this problem uses a 2-dimensional Boolean array, X, with n rows and $W + l$ columns $X[i, j]$, $l \leq I \leq n$, $0 \leq j \leq W$, is TRUE if and only if there is a subset of $\{a_1, a_2, a_3, \ldots, a_n\}$ whose elements sum to j. **[2008]**

Which of the following is valid for $2 \leq i \leq n$ and $a_i \leq j \leq W$?
(a) $X[i, j] = X[i - 1, j] \vee X[i, j - a_i]$
(b) $X[i, j] = X[i - 1, j] \vee X[i - j - a_i]$
(c) $X[i, j] = X[i - 1, j] \wedge X[i, j - a_i]$
(d) $X[i, j] = X[i - 1, j] \wedge X[i - 1, j - a_i]$

Solution: (b)
Use Principle of optimality
Hence, the correct option is (b).

3. The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_l, a_2, a_3, \ldots, a_n\}$, and positive integer W, is there a subset of S whose elements sum to W? A dynamic programme for solving this problem uses a 2-dimensional Boolean array, X, with n rows and $W + l$ columns $X[i, j]$, $l \leq I \leq n, 0 \leq j \leq W$, is TRUE if and only if there is a subset of $\{a_l, a_2, a_3, \ldots, a_n\}$ whose elements sum to j. **[2008]**
Which entry of the array X, if TRUE, implies that there is a subset whose elements sum to W?
(a) $X[l, w]$
(b) $X[n, 0]$
(c) $X[n, W]$
(d) $X[n - l, n]$

Solution: (c)
Hence, the correct option is (c).

4. Consider the following C programme that attempts to locate an element x in an array $Y[\ ]$ using binary search. **[2008]**

The programme is enoreneous

```
f(int Y[10], int x){
  int i, j, k;
  i = 0; j=9;
  do(
     k = (i+j)/2;
    if(Y[k] < x) i = k; else j = k;
    } while ((Y[k]! = x) && (I < j));
  if (Y[k] – x) printf("x is in
  the array");
  else printf("x is not in the
  array");
}
```

On which of the following contents of Y and x does the programme fail?
(a) Y is [1 2 3 4 5 6 7 8 9 10] and $x < 10$
(b) Y is [1 3 5 7 9 11 13 15 17 19] and $x < 1$
(c) Y is [2 2 2 2 2 2 2 2 2 2] and $x > 2$
(d) Y is [2 4 6 8 10 12 14 16 18 20] and $2 < x < 20$ and x is even
In option 'c' ($y[k]! = x$ and $i < j$) always, so it goes into infinite loop

Solution: (a)
Hence, the correct option is (a).

5. Consider the following C program that attempts to locate an element x in an array Y[ ] using binary search. **[2008]**

The program is enoreneous

```
f(int Y[10], int x){
  int i, j, k;
   i = 0; j=9;
   do(
    k = (i+j)/2;
    if(Y[k] < x) i = k; else j = k;
   } while ((Y[k]! = x) && (I < j));
  if (Y[k] – x) printf("x is in
  the array");
  else printf("x is not in the
  array");
}
```

The correction needed in the program to make it work properly is

(a) Change line 6 to if $(Y[k] < x)$ $i = k + l$; else $j = k - 1$;
(b) Change line 6 to if $(Y[k] < x)$ $i = k - 1$; else $j = k + 1$;
(c) Change line 6 to if $(Y[k] <= x)$ $i = k$; else $i = k$;
(d) Change line 7 to while $((Y[k] == x)$ and $(I < j))$;

It will cause the loop to terminate wherein the condition $i < j$ now becomes false.

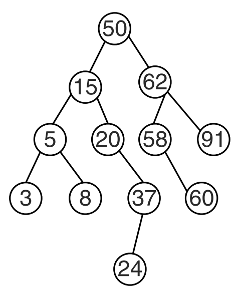
Solution: (c)

Hence, the correct option is (c).

6. A binary search tree is generated by inserting in order the following integers:
50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24
The number of nodes in the left subtree and right subtree of the root respectively is **[1996]**

(a) (4, 7) (b) (7, 4)
(c) (8, 3) (d) (3, 8)

Solution: (b)



Hence, the correct option is (b).

7. Obtain the optimal binary search tree with equal probabilities for the identifier set (a_1, a_2, a_3) = (if, stop, while) **[1991]**

Solution:
Apply the Dynamic Programming formula for obtaining minimum cost Binary Search tree.

$$\text{COST}(i, j) = \min\{c[I, k-1] + c(k, j)] + w(I, j) \quad (1)$$
$$i < k \leq j$$

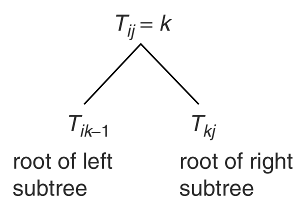
$\text{COST}(I, i) = \Phi$

$$w(i, j) = q_i + \sum_{e=i+1}^{j} (p_c + q_c); w(I, i) = q_i$$

$R(i, i)$ = the value of 'k' that minimise equation (1) is

$R(i, j) = \Phi$; the root of B.S.T

Let $R(i, j)$ be the root of B.S.T T_{ij} then the tree is constructed as



Five-marks Question

1. Fill in the blanks in the following template of an algorithm to compute all pairs shortest path lengths in a directed graph G with $n*n$ adjacency matrix A. $A[i, j]$ equals to 1 if there is an edge in G from i to j, and 0 otherwise. Your aim in filling in the blanks is to ensure that the algorithm is correct.

```
INITIALIZATION: For i = 1 ... n
{For j = 1 ... n
{ if A[i, j]=0 then P[i, j] = ___
  else P[i, j] =____;}
ALGORITHM: For i = 1 ...n
{For j = 1 ...n
{For k = 1 ...n
{P[_,___]= min{_______,_______};}
}
}
```

(a) Copy the complete line containing the blanks in the Initialization step and fill in the blanks.
(b) Copy the complete line containing the blanks in the Algorithm step and fill in the blanks.
(c) Fill in the blank: The running time of the Algorithm is O (____). **[2002]**

Solution:

```
for i ← 1 to n
for j ← 1 to n
if (A[i, j] = Φ) then P[i, j] = Φ;
else P(i, j] = 1;
for i ← 1 to n
for j ← 1 to n
for k ← 1 to n
P [i, j] = min{P[i, k]+P[k, j],P[i,
  j]}
```

Time complexity is $O(n^3)$

Chapter 5

P and NP Concepts

One-mark Questions

1. Let G be a graph with n vertices and m edges. What is the tightest upper bound on the running time of Depth First Search on G, When G is represented as an adjacency matrix? **[2014]**

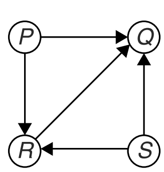
(a) $\Theta(n)$ (b) $\Theta(n + m)$
(c) $\Theta(n^2)$ (d) $\Theta(m^2)$

Solution: (c)
If the graph is represented using adjacency matrix, then total number of elements matrix are bounded by $\theta\,(n^2)$. So DFS algorithm takes $\theta\,(n^2)$.
Hence, the correct option is (c).

2. Consider a rooted n node binary tree represented using pointers. The best upper bound on the time required to determine the number of subtrees having exactly 4 nodes is $O(n^a \log^b n)$. Then the value of $a + 10b$ is _______. **[2014]**

Solution: 1
The best algorithm on the time required to determine the number of subtrees having exactly 4 nodes takes $\theta\,(n)$
Hence, $a = 1$ and $b = 0$, so $a + 10(b) = 1$

3. Consider the directed graph given below. **[2014]**



Which one of the following is TRUE?

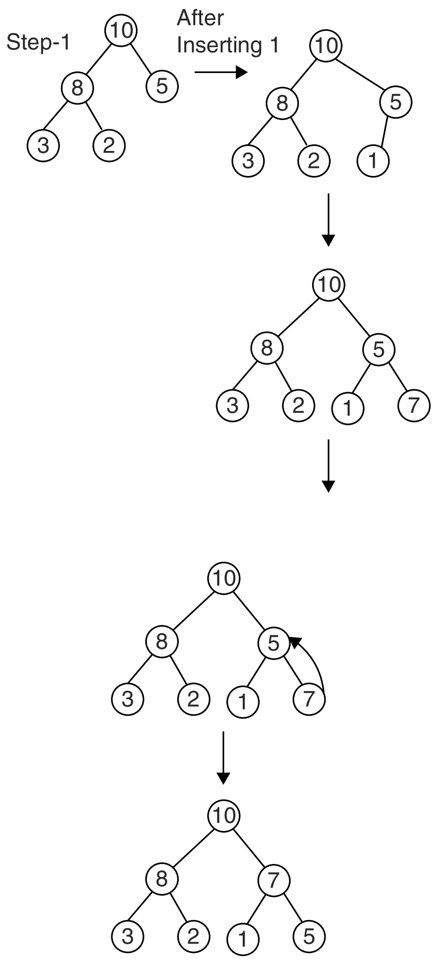
(a) The graph does not have any topological ordering.
(b) Both *PQRS* and *SRQP* are topological orderings.
(c) Both *PSRQ* and *SPRQ* are topological orderings.
(d) *PSRQ* is the only topological ordering.

Solution: (c)
To get topological ordering, we have to start from the vertex of degree 0, and add that vertex to the topological ordering sequence and then remove it together with all its edges from the graph. Continue this process to remaining all vertices.
Hence, there are two topological ordering are in the graph *PSRQ* and *SPRQ*.
Hence, the correct option is (c).

4. A priority queue is implemented as a Max-leap. Initially, it has five elements. The level-order traversal of the heap is: 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the leap in that order. The level-order traversal of the heap after the insertion of the elements is: **[2014]**

(a) 10, 8, 7, 3, 2, 1, 5
(b) 10, 8, 7, 2, 3, 1, 5
(c) 10, 8, 7, 1, 2, 3, 5
(d) 10, 8, 7, 5, 3, 2, 1

Solution: (a)



Level order traversal: 10, 8, 7, 3, 2, 1, 5
Hence, the correct option is (a).

5. Consider the tree arcs of a BFS traversal from source node *W* in an unweighted, connected, indirected graph. The tree *T* formed by the three arcs is a data structure for computing **[2014]**

(a) The shortest path between every pair of vertices.
(b) The shortest path from ***W*** to every vertex in the graph.
(c) The shortest paths from ***W*** to only those nodes that are leaves of ***T***.
(d) The longest path in the graph.

Solution: JFS algorithm is useful for finding, shortest path distance from source node to every either node.

6. Which of the following statements are TRUE? **[2013]**

1. The problem of determining whether there exists a cycle in an undirected graph is in *P*.
2. The problem of determining whether there exists a cycle in an undirected graph in *NP*.
3. If a problem *A* is *NP*-complete, there exit a non-deterministic polynomial time algorithm to solve *A*.

(a) 1, 2 and 3
(b) 1 and 2 only
(c) 2 and 3 only
(d) 1 and 3 only

Solution: (a)
Hence, the correct option is (a).

7. Let *A* be a problem that belongs to the class *NP*. Then which one of the following is TRUE? **[2009]**

(a) There is no polynomial time algorithm for *A*.
(b) If *A* can be solved deterministically in polynomial time, then *P* = *NP*.
(c) If *A* is *NP*-Hard, then it is *NP*-complete.
(d) *A* may be undecidable.

Solution: (c)
By definition of *NP*-Hard and *NP*-complete rest all choices can be shown to be false.
If *A* is *NP*-Hard, then it is *NP*-complete.
Hence, the correct option is (c).

8. Let *S* be an *NP*-complete problem and *Q* and *R* be two other problems not known to be in *NP*. *Q* is polynomial time reducible to *S* and *S* is polynomial-time reducible to *R*. Which one of the following statements is true? **[2006]**

(a) *R* is *NP*-complete
(b) *R* is *NP*-Hard
(c) *Q* is *NP*-complete
(d) *Q* is *NP*-Hard

Solution: (b)
Since $S \alpha R$ '*R*' is definitely *NP*-Hard *R* is *NP*-Hard.
Hence, the correct option is (b).

9. The problem 3-SAT and 2-SAT are **[2004]**

(a) both in *P*
(b) both *NP*-complete
(c) *NP*-complete and in *P* respectively
(d) undecidable and *NP*-complete respectively

Solution: (c)

For SAT problem, 1-SAT and 2-SAT are considered to be in P from 3-SAT onwards, it is treated to be Hard.

3-SAT is *NP*-Complete. So it is *NP*, *NP*-Hard, and *NP*-complete.

2-SAT is solvable in poly-time. So it is *P*, *NP*, and Co*NP*.

Hence, the correct option is (c).

10. Ram and Shyam have been asked to show that a certain problem Π is *NP*-complete. Ram shows a polynomial time reduction from the 3-SAT problem to Π and Shyam shows a polynomial time reduction from Π to 3-SAT. Which of the following can be inferred from these reductions? **[2003]**

(a) Π is *NP*-Hard but not *NP*-complete
(b) Π is in *NP*, but is not *NP*-complete
(c) Π is *NP*-complete
(d) Π is neither *NP*-Hard, nor in *NP*

Solution: (a)

For a problem to be *NP*-complete, Π should belong to the class *NP*.

Hence, the correct option is (a).

Two-marks Questions

1. Suppose we have a balanced binary search tree T holding n numbers. We are given two numbers L and H and wish to sum up all the numbers in T that lie between L and H. Suppose there are m such numbers in T. If the tightest upper bound on the time to compute the sum is $O(n^a \log^b n + m^c \log^d n)$, the value of $a + 10b + 100c + 1000d$ is_______. **[2014]**

Solution: 110

The total time required to perform given task is $O(\log n + m)$ so substitute $a = 0$, $b = 1$, $c = 1$ and $d = 0$

Therefore, $a + 10b + 100c + 1000d = 110$

2. The subset-sum problem is defined as follows: Given a set S of n positive integers and a positive integer W, determine whether there is a subset of S whose elements sum to W. An algorithm Q solves this problem in O (nW) time. Which of the following statements is false? **[2008]**

(a) Q solves the subset-sum problem in polynomial time when the input is encoded in unary
(b) Q solves the subset-sum problem in polynomial time when the input is encoded in binary
(c) The subset sum problem belongs to the class *NP*
(d) The subset sum problem is *NP*-Hard

Solution: (b)

Sum of subsets problem is *NP*-Hard and it also belongs to the class of *NP*. It may have linear time algorithm when the *I*/*P* is encoded in unary system.

3. Let $SHAM_3$ be the problem of finding a Hamiltonian cycle in a graph $G = (V, E)$ with $|V|$ divisible by 3 and $DHAM_3$ be the problem of determining if a Hamiltonian cycle exists in such graphs. Which one of the following is true? **[2006]**

(a) Both $DHAM_3$ and $SHAM_3$ are *NP*-Hard
(b) $SHAM_3$ is *NP*-Hard, but $DHAM_3$ is not
(c) $DHAM_3$ is *NP*-Hard, but $SHAM_3$ is not
(d) Neither $DHAM_3$ nor $SHAM_3$ is *NP*-Hard

Solution: (a)

Hence, the correct option is (a).

4. Consider three decision problems P_1, P_2 and P_3. It is known that P_1 is decidable and P_2 is undecidable. Which one of the following is TRUE? **[2005]**

(a) P_3 is decidable if P_1 is reducible to P_3
(b) P_3 is undecidable if P_3 is reducible to P_2
(c) P_3 is undecidable if P_2 is reducible to P_3
(d) P_3 is decidable if P_3 is reducible to P_2's complement

Solution: (b)

Using the concept of polynomial time reduction

Hence, the correct option is (b).

5. Which of the following problems is not *NP*-Hard? **[1992]**

(a) Hamiltonian circuit problem
(b) The $\frac{0}{1}$ Knapsack problem
(c) Finding bi-connected components of a graph
(d) The graph colouring problem

Solution: (c)

Hamiltonian circuit, $\frac{0}{1}$ Knapsack and graph colouring are also hard problems as they do not have a polynomial time complexity algorithm, However finding bi-connected components of a graph is a problem in the class of 'p'.

Hence, the correct option is (c).

Chapter 6

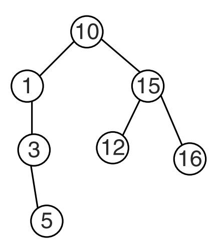
Optimal Binary Search Tree

One-mark Questions

1. The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree (the height is the maximum distance of a leaf node from the root)? **[2004]**

(a) 2 (b) 3
(c) 4 (d) 6

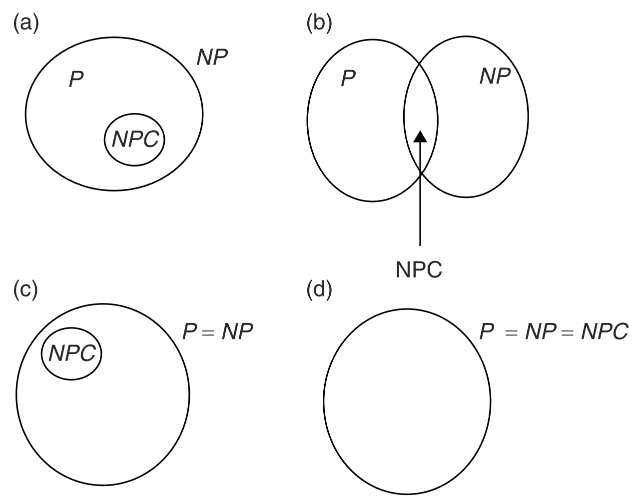
Solution: (b)
The leaf '5' is at a distance of path length 3 from the root.
It's binary search tree will be:



Hence, the correct option is (b).

Two-mark Questions

1. Suppose a polynomial time algorithm is discovered that correctly computes the largest clique in a given graph. In this scenario, which one of the following represents the correct Venn diagram of the complexity classes *P*, *NP* and *NP*-complete (*NPC*)? **[2014]**



Solution: (d)
Computing largest Clique in a given graph, is *NP*-Hard problem. So if a polynomial time algorithm discovered for *NP*-Hard problem then we can discover polynomial time algorithm for every *NP*-Hard problem i.e. $P = NPH$ and $P = NP$.
Therefore, $P = NPC$
Hence, $P = NP = NPC$
Hence, the correct option is (d).

2. Consider the decision problem 2CNFSAT defined as follows:
$\{\Phi \mid \Phi$ is a satisfiable propositional formula in CNF with at most two literals per clause$\}$
For example, $\Phi = (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_4)$ is a Boolean formula and it is in 2CNFSAT.
The decision problem 2CNFSAT is **[2014]**

(a) *NP*-complete
(b) Solvable in polynomial time by reduction to directed graph reachability.
(c) Solvable in constant time since any input instance is satisfiable.
(d) *NP*-Hard, but not *NP*-complete.

Solution: (b)
2CNFSAT problem is *P*-class problem so it can be solved in polynomial time by reduction to polynomial time reachability.

3. A programme takes as input a balanced binary search tree with n leaf-nodes and computes the value of a function $g(x)$ for each node x. If the cost of computing $g(x)$ is min (number of leaf-nodes in left-subtree of x, number of leaf-nodes in right-subtree of x) then the worst-case time complexity of the programme is **[2004]**

(a) $\Theta(n)$ (b) $\Theta(n\log n)$
(c) $\Theta(n^2)$ (d) $\Theta(n^2\log n)$

Solution: (a)
For the nodes at leaf level, the number of nodes = n

The next level it is $\frac{n}{2}$, then next $\frac{n^2}{2}, \ldots \frac{n^k}{2}$.

For nodes at leaf level number of leaves = 0
For nodes at leaf next level number of leaves = 1

i.e. $n \ldots 0 \;\; \frac{n}{2} - 1 \;\; \frac{n}{4} - 2 \;\; \frac{n^k}{2} - K$

Therefore total time

$$= \sum_{i=1}^{k/2^k} \left(\frac{n}{2^i}\right) \cdot i$$

$$= n\sum_{i=1}^{k/2^k} \left(\frac{n}{2^i}\right)$$

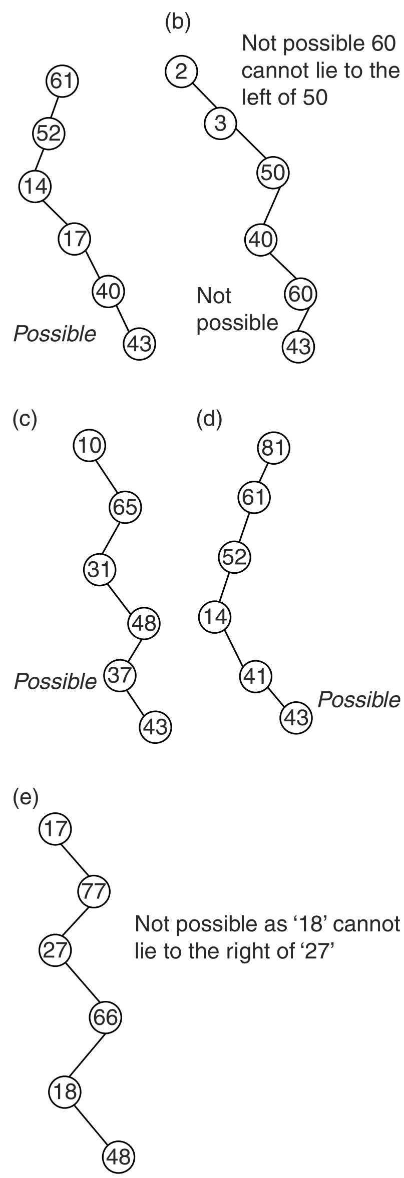
$$= O(n)$$

Hence, the correct option is (a).

4. A binary search tree is used to locate the number 43. Which of the following probe sequences are possible and which are not? Explain. **[1996]**

(a) 61 52 14 17 40 43
(b) 2 3 50 40 60 43
(c) 10 65 31 48 37 43
(d) 81 61 52 14 41 43
(e) 17 77 27 66 18 43

Solution: (a)



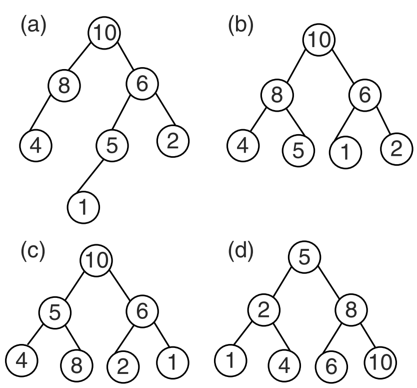
Hence, the correct option is (a).

Chapter 7

Miscellaneous Topics

One-mark Questions

1. Max heap is a heap where the value of each parent is greater than or equal to the value of children. Which of the following is a max-p? **[2011]**



Solution: (b)
Trees in (e) and (d) violate the property of Max Heap. Tree (a) satisfies the property of Max Heap, but it is not a complete Binary Tree. Tree (b) is the correct answer. (a) is not a complete binary tree. In (c) and (d), heap property is not satisfied between 5 and 8.
Hence, the correct option is (b).

2. What is the number of swaps required to sort n-elements using selection sort, in the worst case? **[2009]**

(a) $\Theta(n)$ (b) $\Theta(n\log n)$
(c) $\Theta(n^2)$ (d) $\Theta(n^2\log n)$

Solution: (a)
To sort 'n' elements using selection sort it requires $O(n)$ swaps.
Hence, the correct option is (a).

3. Which one of the following array represents a max heap? **[2009]**

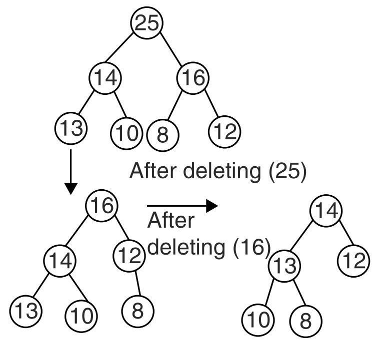
(a) {25, 12, 16, 13, 10, 8, 14}
(b) {25, 14, 13, 16, 10, 8, 12}
(c) {25, 14, 16, 13, 10, 8, 12}
(d) {25, 14, 12, 13, 10, 8, 16}

Solution: (c)
In option (a) (13) cannot be the child of (12) in max Heap in option (b) (16) cannot be the child of (14). In option (d) (16) cannot be the child of smaller value node.
In (a), $s[3]$ which is the left child of $a[1]$ is greater than the parent (13 > 12). In (b), also $a[3] > a[1]$ (16 > 14). In (d), $a[6]$ which is right child of $a[2]$ is greater than $a[2]$ (16 > 12).
Hence, the correct option is (c).

4. It is the content of the array after two operations on the correct answer to the previous question? **[2009]**

(a) {14, 13, 12, 10, 8}
(b) {14, 12, 13, 8, 10}
(c) {14, 13, 8, 12, 10}
(d) {14, 13, 12, 8, 10}

Solution: (d)

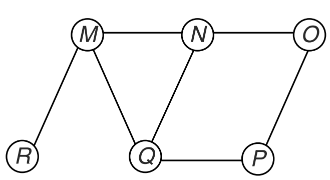


The height of a Max Heap is Θ (log n). While insertion, we need to traverse from leaf element to root (in worst). If we perform binary search to find the correct position then we need to do Θ (log log n) comparisons. In reality, it is not possible to perform binary search on elements from leaf to root as they are not in sequence.
Hence, the correct option is (d).

5. The most efficient algorithm for finding the number of connected components in an undirected graph on n vertices and m edges has time complexity **[2008]**
(a) $\Theta(n)$
(b) $\Theta(m)$
(c) $\Theta(m + n)$
(d) $\Theta(mn)$

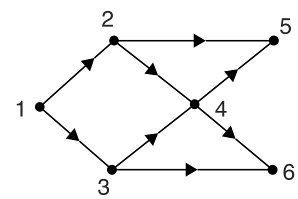
Solution: (c)
This is the case when the graph is represented by cost Adjacency matrix.
Hence, the correct option is (c).

6. The Breadth-first search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is **[2008]**
(a) *MNOPQR*
(b) *NQMPOR*
(c) *QMNPRO*
(d) *QMNPOR*



Solution: (c)
The remaining choices violates the FIFO discipline of the queue and hence are not valid BFS traversals.
Hence, the correct option is (c).

7. Consider the DAG with $V = \{1, 2, 3, 4, 5, 6\}$ is shown below. Which of the following is NOT a topological ordering? **[2007]**



(a) 1 2 3 4 5 6
(b) 1 3 2 4 5 6
(c) 1 3 2 4 6 5
(d) 3 2 4 1 6 5

Solution: (d)
Topological sort/ordering starts with that vertex which does not have any precedence. In the given graph, vertex ‘3’ has a precedence from vertex ‘1’ So, it is incorrect.
Hence, the correct option is (d).

8. Which of the following sorting algorithms has the lowest worst case complexity? **[2007]**
(a) Merge sort
(b) Bubble sort
(c) Quick sort
(d) Selection sort

Solution: (a)
Merge sort takes a time of $O(n \log n)$ in all cases of input. Whereas other sorting techniques have complexity of $O(n^2)$ in worst case.
Hence, the correct option is (a).

9. In a binary max heap containing n numbers, the smallest element can be found in time **[2006]**
(a) $\Theta(n)$
(b) $\Theta(\log n)$
(c) $\Theta(\log \log n)$
(d) $\Theta(l)$

Solution: (a)
Smallest element lie at the leaf level which has roughly $\frac{n}{2}$ elements that would require number ofcomparison and number of elements.
Hence, the correct option is (a).

10. Which one of the following in place sorting algorithms needs the minimum number of swaps? **[2006]**

(a) Quick sort
(b) Insertion sort
(c) Selection sort
(d) Heap sort

Solution: (c)
In selection sort, each iteration takes one swap in the worst case. Hence it requires $O(n)$ swaps in the worst case.
Hence, the correct option is (c).

11. An element in an array X is called a leader if it is greater than all elements to the right of it in X. The best algorithm to find all leaders in an array **[2006]**

(a) Solves it in linear time using a left to right pass of the array
(b) Solves it in linear time using a right to left pass of the array
(c) Solves it using divide and conquer in time $9(n \log n)$
(d) Solves it in time $\Theta(n^2)$

Solution: (b)
Algorithm to determine leaders in an array: int max = a[n]

```
for i ← n – 1 to 1 by –1
{
if(a[i] > max)
{
print a[i];
max = a[i];
}
}
```

While scanning the array from right to left remember the greatest element seen so far and compare it with the current element to test for leadership.
Hence, the correct option is (b).

12. The tightest lower bound on the number of comparisons, m the worst case, for comparison based sorting is of the order of **[2004]**

(a) n
(b) n^2
(c) $n \log n$
(d) $n \log^2 n$

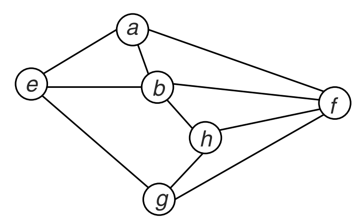
Solution: (c)
In comparison based algorithm, tightest lower bound is $O(n \log n)$ and tightest upper bound if $O(n^2)$
Hence, the correct option is (c).

13. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree? **[2003]**

(a) 7 5 1 0 3 2 4 6 8 9
(b) 0 2 4 3 1 6 5 9 8 7
(c) 0 1 2 3 4 5 6 7 8 9
(d) 9 8 6 4 2 3 0 1 5 7

Solution: (c)
In-order traversal when carried out on Binary Search Tree, results in a sorted list.
Hence, the correct option is (c).

14. Consider the following graph



Among the following sequences

I. *a b e g h f*
II. *a b f e h g*
III. *a b f h g e*
IV. *a f g h b e*

Which are depth-first traversals of the above graph? **[2003]**

(a) I, II and IV only
(b) I and V only
(c) II, III and IV only
(d) I, III and IV only

Solution: (d)
a, b, f, e, h, g is not possible as one cannot visit '*e*' after '*f*'
Hence, the correct option is (d).

15. In a heap with n-elements with the smallest element at the root, the 7th smallest element can be found in time **[2003]**

(a) $\Theta(n \log n)$ (b) $\Theta(n)$
(c) $\Theta(\log n)$ (d) $\Theta(l)$

Solution: (c)
In-order to find 7th smallest element, we have to perform '7' deletion operation so it takes $O(7$

$\log n) = O(\log n)$
Hence, the correct option is (c).

16. Consider any array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the array ($i <= n$), the index of the parent is **[2001]**

(a) $i - l$ (b) $\left\lfloor \frac{i}{2} \right\rfloor$
(c) $\left\lceil \frac{i}{2} \right\rceil$ (d) $\left\lceil \frac{i+l}{2} \right\rceil$

Solution: (b)
In the array representation of binary tree, the parent is at location $[\frac{i}{2}]$, whereas left child is at $2i$ and right child at $2i + l$.
Hence, the correct option is (b).

17. Suppose we want to arrange the n numbers stored in any array such that all negative values occur before all positive ones. Minimum number of exchanges required in the worst case is **[1999]**
(a) $n - l$ (b) n
(c) $n + 1$ (d) None

Solution: (a)
Worst case input is when the negative number occur at the end of positive number's minimum number of comparisons in such a case is $(n - 1)$
Hence, the correct option is (a).

18. The number of articulation points of the following graph is **[1999]**
(a) 0 (b) 1 (c) 2 (d) 3

Solution: (d)
Nodes 2, 3, 5 are the Articulation points.
Hence, the correct option is (d).

19. Give the correct matching for the following pairs:
(a) O (log n) (P) Selection
(b) O (n) (Q) Insertion sort
(c) O (n log n) (R) Binary search
(d) O (n^2) (S) Merge sort **[1998]**

(a) *A-RB-PC-QD-S*
(b) *A-RB-QC-SD-P*
(c) *A-PB-RC-SD-Q*
(d) *A-PB-SC-RD-Q*

Solution: (b)
Binary search – O (log$n$)
Selection sort – O (n^2)
Insertion sort – O (n) (or) O (n^2)
Merge sort – O ($n$log$n$)
Hence, the correct option is (b).

20. The correct matching for the following pairs is
(a) All pairs shortest paths
(b) Quick Sort
(c) Minimum weight spanning tree
(d) Connected Components
(1) Greedy
(2) Depth-first search
(3) Dynamic Programming
(4) Divide and Conquer [1997]
(a) A–2, B–4, C–1,D–3
(b) A–3, B–4, C–1, D–2
(c) A–3, B–4, C–2, D–1
(d) A–4, B–1, C–2, D–3

Solution: (b)
Hence, the correct option is (b).

Two-marks Questions

1. We have a binary heap on n-elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is **[2008]**
(a) $\Theta(\log n)$
(b) $\Theta(n)$
(c) $\Theta(n \log n)$
(d) $\Theta(n^2)$

Solution: (b)
Procedure Heapify with adjust would require time of O (n) for n-elements, now with addition n-elements, total being '$2n$' would still be order of O (n). We can reduce the problem to building Heap for $2n$ elements. Time complexity for building heap is O (n).
Hence, the correct option is (b).

2. In the following C function, let $n >= m$.

```
int gcd(n, m)
{
if (n % m = = 0) return m;
n = n % m;
return gcd (m, n);
}
```

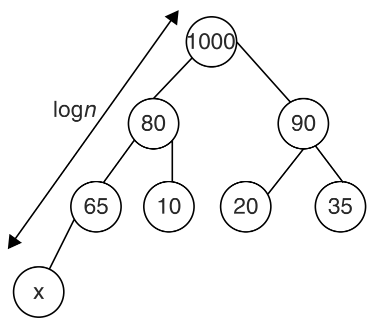
How many recursive calls are made by this function? **[2007]**

(a) $\Theta(\log_2 n)$
(b) $\Omega(n)$
(c) $\Theta(\log_2 \log_2 n)$
(d) $\Theta(\sqrt{n})$

Solution: (c)
Hence, the correct option is (c).

3. Consider the process of inserting an element into a Max Heap, where the Max Heap is represented by an array. Suppose we perform a binary search on the path from the new leaf to the root to find the position for the newly inserted element, the number of comparisons performed is: **[2007]**
(a) $\Theta(\log_2 n)$
(b) $\Theta(\log_2 \log_2 n)$
(c) $\Theta(n)$
(d) $\Theta(n \log 2\ n)$

Solution: (b)
Binary search to be applied along the path which has $\log n$ elements.



The height of a Max Heap is $\Theta(\log n)$. While insertion, we need to traverse from leaf element to root (in worst). If we perform binary search for finding the correct position then we need to do $\Theta(\log \log n)$ comparisons. In reality, it is not possible to perform binary search on elements from leaf to root as they are not in sequence.

4. Let T be a depth-first search tree in an undirected graph G. Vertices u and v are leaves of this tree T. The degrees of both u and v in G are at least 2. Which one of the following statements is true? **[2006]**
(a) There must exist a vertex w adjacent to both u and v in G
(b) There must exist a vertex w whose removal disconnects u and v in G
(c) There must exist a cycle in G containing u and v
(d) There must exist a cycle in G containing u and all its neighbours in G.

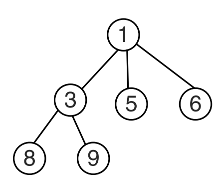
Solution: (d)
Hence, the correct option is (d).

5. Given two arrays of numbers $a_1,\ldots a_n ,\ldots,$ and $b_i,\ldots b_n$ where each number is 0 or 1, the fastest algorithm to find the largest span (i, j) such that $a_i + a_i + i\ldots aj = b_i + b_i + i\ldots bj$ or report that there is not such span, **[2006]**
(a) Takes $O(3^n)$ and $\Omega(2^n)$ time of hashing is permitted
(b) Takes $O(n3)$ and $\Omega\ (n^{2.5})$ time in the key comparison model
(c) Takes $\Theta\ (n)$ time and space
(d) Takes $O(Vn)$ time only if the sum of the $2n$ elements is an even number

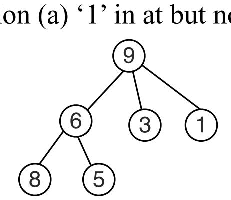
Solution: (c)
Hence, the correct option is (c).

6. A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-heap can be represented by an array as follows: root is stored in the first location, a [0], nodes in next level, from left to right, is stored from a[*l*] [3]. The nodes from the second level of the tree from the left to right are stored from a[4] location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location a[n] and pushing it up the tree to satisfy the heap property. **[2006]**
Which one of the following is a valid sequence of elements in an array representing 3-ary max heap?
(a) 1, 3, 5, 6, 8, 9
(b) 9, 6, 3, 1, 8, 5
(c) 9, 3, 6, 8, 5, 1
(d) 9, 5, 6, 8, 3, 1

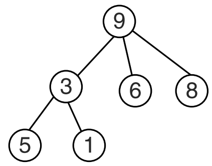
Solution: (d)



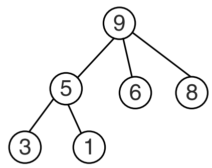
Tree for option (a) '1' in at but not possible



Tree for option (b) ‘8’ is greater than ‘6’ not possible



Tree for option (c) ‘5’ is greater than ‘3’ and hence not Possible



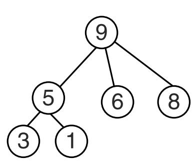
Tree for option (d) is max Heap
Hence, the correct option is (d).

7. A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-heap can be represented by an array as follows: root is stored in the first location, $a[0]$, nodes in next level, from left to right, is stored from $a[1]$ [3]. The nodes from the second level of the tree from the left to right are stored from $a[4]$ location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location a[n] and pushing it up the tree to satisfy the heap property. **[2006]**

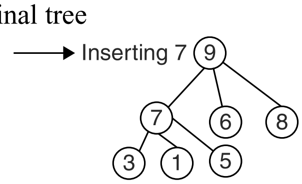
Suppose the elements 7, 2, 10 and 4 are inserted, in that order, into the valid 3-array max heap found in the question 6. Which one of the following is the sequence of items in the array representing the resultant heap?

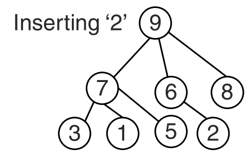
(a) 10, 7, 9, 8, 3, 1, 5, 2, 6, 4
(b) 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
(c) 10, 9, 4, 5, 7, 6, 8, 2, 1, 3
(d) 10, 8, 6, 9, 7, 2, 3, 4, 1, 5

Solution: (a)

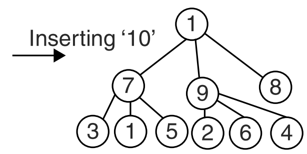


(a) Original tree





In the array it is: 10, 7, 9, 8, 3, 1, 5, 2, 6, 4



Hence, the correct option is (a).

8. Let $G(V, E)$ be an undirected graph with positive edge weights. Dijkstra’s single source shortest path algorithm can be implemented using the binary heap data structure with time complexity **[2005]**

(a) $O(|V^2|)$
(b) $O((|E|+|V| \log |V|)$
(c) $O(V |\log| V.J)$
(d) $O(E. |V| \log |V|)$

Solution: (b)
Using Binary heap Data structure, the time complexity is $O((E + V) \log V)$
Hence, the correct option is (b).

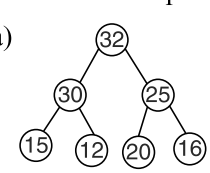
9. Suppose there are [log n] sorted lists of n log n-elements each. The time complexity of producing a sorted list of all these elements is: (Hint: Use a heap data structure) **[2005]**

(a) $O(n \log \log n)$
(b) $O(n \log n)$
(c) $\Omega(n \log n)$
(d) $\Omega(n^{3/2})$

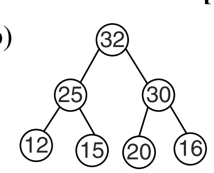
Solution: (b)
Total number of elements in all the lists log $n \cdot \dfrac{n}{\log n} = n.$ A heap of ‘n’ elements and sorting it require $O(n \log n)$
Hence, the correct option is (b).

10. The elements 32, 15, 20, 30, 12, 25, 16, are inserted one by one in the given order into a max Heap. The resultant max Heap is **[2004]**

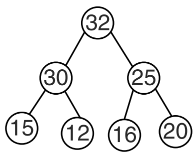
(a)



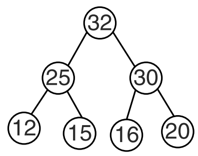
(b)



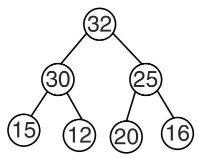
(c)



(d)



Solution: (a)

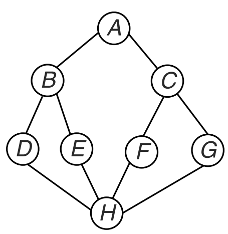


Hence, the correct option is (a).

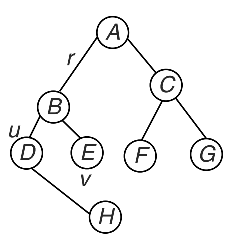
11. Consider an undirected unweighted graph G. Let a breadth-first traversal of G be done starting from a node r. Let $d(r, u)$ and $d(r, v)$ be the lengths of the shortest paths from r to u and v respectively in G. If u is visited before during the breadth-first traversal which of the following statements is correct **[2001]**

(a) $d(r, u) < d(r, v)$
(b) $d(r, u) > d(r, v)$
(c) $d(r, u) \leq d(r, v)$
(d) None of the above

Solution: (c)
If 'u' is visited before 'v', then the distance from 'r' to 'u' will be less than or equal to the distance from 'r' to 'v' as shown in the graph below



(G)



(B.T.T)

Hence, the correct option is (c).

12. How many undirected graphs (not necessarily connected) can be constructed out of a given set $V = \{v_1, v_2, \ldots v_n\}$ of n vertices? **[2001]**

(a) $\frac{n(n-1)}{2}$
(b) 2^n
(c) $n!$
(d) $2n^{(n-1)/2}$

Solution: (d)
It is based on Kirchoff's Theorem
Hence, the correct option is (d).

13. Lt G be an undirected graph. Consider a depth-first traversal of G, and let T be the resulting depth-first search tree. Let u be a vertex in G and let v be the first new (unvisited) vertex visited after visiting u in the traversal. Which of the following statements is always true? **[2000]**

(a) $\{u, v\}$ must be an edge in G, and u is a descendant of v in T
(b) $\{u, v\}$ must be an edge in G, and v is a descendant of u in T
(c) If $\{u, v\}$ is not an edge in G then u is a leaf in T
(d) If $\{u, v\}$ is not an edge in G then u and v must have the same parent in T

Solution: (c)
(c) is always true but (b) and (d) are true in only few cases, (a) is never true.
Hence, the correct option is (c).

14. If $Tl = O(1)$, give the correct matching for the following pairs:

List I	List II
(M) $T_n = T_{n-1} + n$	(U) $T_n = O(n)$
(N) $T_n = T_{n/2} + n$	(V) $T_n = O(n\log n)$
(O) $T_n = T_{n/2} + n\log n$	(W) $T_n = O(n^2)$
(P) $T_n = T_{n-1} + \log n$	(X) $T_n = O(\log n!)$

[1999]

(a) M–W, N–V, O–U, P–X
(b) M–W, N–U, O–X, P–V
(c) M–V, N–W, O–X, P–U
(d) M–W, N–U, O–V, P–X

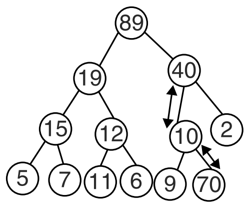
Solution: M–W, N–U, O–V, P–X

15. The minimum number of interchanges needed to convert the array 89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70 into a heap with the maximum element at the root is

[1996]

(a) 0 (b) 1 (c) 2 (d) 3

Solution: (c)
Tree Representation of The array



Hence, the correct option is (c).

16. The average number of key comparisons done on a successful sequential search in list of length n is **[1996]**

(a) $\log n$ (b) $\frac{n-1}{2}$

(c) $\frac{n}{2}$ (d) $\frac{n+1}{2}$

Solution: (d)

If x is found at loc '1' $\rightarrow$ 1 comparison
x is found at loc '2' $\rightarrow$ 2 comparison
x is found at loc '3' $\rightarrow$ 3 comparison
x is found at loc 'n' $\rightarrow$ 'n' comparison

$$\text{Total comparison} = \frac{n(n+1)}{2}$$

$$\text{Therefore average} = \frac{\frac{n(n+1)}{2}}{n}$$

Hence, the correct option is (c).

17. Which one of the following statements is false? **[1994]**

(a) Optimal binary search tree construction can be performed efficiently using dynamic programming.
(b) Breadth-first search cannot be used to find connected components of a graph.
(c) Given the prefix and postfix walks over a binary tree, the binary tree cannot be uniquely constructed.
(d) Depth-first search can be used to find connected components of a graph.

Solution: (b)

Breadth-first search can be used to find connected components of a graph.
Hence, the correct option is (b).

18. Following algorithm(s) can be used to sort n integers in the range $[1\ldots n^3]$ in $0(n)$ time **[1992]**

(a) Heap sort
(b) Quick sort
(c) Merge sort
(d) Radix sort

Solution: (d)

Radix Sort is a mechanical sorting technique. Remaining all are comparison based sorting whose lower bound complexity is $0(n \log n)$
Hence, the correct option is (d).

19. The minimum number of comparisons required to sort five elements is **[1991]**

Solution:

Minimum may be four, using bubble sort (or) selection sort (or) insertion sort

20. Match the pairs in the following: **[1990]**

List I	List II
(a) Straseen's matrix multiplication algorithm	(p) Greedy method
(b) Kruskal's minimum spanning tree algorithm	(q) Dynamic programming
(c) Biconnected components algorithm	(r) Divide and Conquer
(d) Floyd's shortest path algorithm	(s) Depth-first search

Solution: a–r, b–p, c–s, d–q

21. Match the pairs in the following **[1990]**

List I	List II
(a) Heap construction	(p) $O(n \log 10n)$
(b) Constructing hash table with linear probing	(q) $O(n)$
(c) AVL Tree construction	(r) $O(n^2)$
(d) Digital tree construction	(s) $O(n \log_2 n)$

Solution: a–q, b–r, c–s, d–p

Five-marks Questions

1. An array A contains $n \geq 1$ positive integers in the locations $A[1], A[2], \ldots, A[n]$.
The following programme fragment prints the length of a shortest sequence of consecutive elements of A, $A[i], A[i+1], \ldots, A[j]$ such that the sum of their values is $\geq M$, a given positive number. It prints '$n + 1$' if no such sequence exists. Complete the programme by filling the boxes. In each case use the simplest possible expression. Write only the line number and the contents of the box.

```
 1. begin
 2. i: + 1j: = 1;
 3. sum := _
 4. min: = n; finish: = false;
 5. while not finish do
 6. if _ then
 7. if j = n then finish: = true.
 8. else
 9. begin
10. j:= j + 1;
11. sum:=; ________
12. end
13. else
14. begin
15. if(j − I) < min then min:=j − 1;
16. sum:= sum − A[i];
17. i:= i + 1;
18. end
19. writeIn (min + 1);
20. end
```

[1997]

Solution:
Line # 3: sum $= A\,[j]$
Line # 6: if (sum $< M$)
Line # 11: sum $=$ sum $+ A[j]$

2. A two dimensional array $A\ [l - n]\ [1 - n]$ of integers is partially sorted if
$\forall i\, j\ \varepsilon[1 - n - l]\ A[i]\ [j] < A[i]\ [j+1]$ and $A[i]\ [j] < A\ [i+1]\ [j]$

(a) The smallest item in the array is at $A[i]\ [j]$ where $i = -$ and j $= -(1)$

(b) The smallest item is deleted. Complete the following $O(n)$ procedure to insert item x (which is guaranteed to be smaller than any item in the last row or column) still keeping A partially sorted.

```
Procedure insert (x integer)
var j' j: integer;
begin
(1) i: = l; j:= A [i] [j]: = x;
(2) while (x > or x >) do
(3) if A [i + 1] [j] n A [i] [j] then
      begin
(4) A[i] [j]: = A [i + 1] [j]; i:= i
     + 1
(5) end
(6) else begin
(7)
(8) end
(9) A [i] [j]:= end
```

[1996]

Solution:

(a) $i = l; j = 1$

(b) line 2; while $(x > A[I + j]\ [j]$ or $x > A[i][j + l])$
line 7; $A[i]\ [j] = A[i]\ D + l];$
$j = j + i;$
line 9; $A[i]\ [j] = x;$

3. An array A contains n integers in non-decreasing order, $A[1] \leq A\ [2] \leq \ldots \leq A[n]$. Describe, using Pascal-like pseudo code, a linear time algorithm to find i, j such that $A[i] + A[j] =$ a given integer M, if such i, j exist. **[1994]**

Solution:
1. $i = l;$
2. $j = n;$
3. if $[(A\ [i] + A[j]) = m]$ then print (i, j); exit
4. if $[(A\ [i] + A[j] < m]$ then $i = i + l;$

goto step 3; else
$j = j - 1$; go to step 3;
The complexity of the above algorithm is $O(n)$;

4. The following Pascal programme segments finds the largest number in a two dimensional integer array A $[0 \ldots n - 1, 0 \ldots n - 1]$ using a single loop. Fill up the boxes to complete the programme and write against A, B, C and D in your answer book. Assume that max is a variable to store the largest value and i, j are the indices to the array.

```
Begin max:= is A, i: = 0, j:= 0;
while B do begin
if A [i, j] > max then max: = A[i, j],
```

```
if [C] then j:= j + l else begin
j:=0;
i: =[D]
end
end
end
```
[1993]

Solution: max = Φ

While ($i <$ $\underline{n}$)

if ($j < n - l$)

$i = i + l$

5. Consider the recursive algorithm given below:

```
Procedure bubblesort (n);
var i, j index; temp: item;
if (A[i] > A [i + l])then begin
temp : A[i];
A[i]:= A[i + l];
A[i + l]:= temp
end;
bubblesort (n - 1)
end
```

Let n be the number of times the 'if…then….' Statement gets executed when the algorithm is run with value n. Set up the recurrence relation by defining a_n in terms of a_{n-1} Solve for a_n. **[1993]**

Solution: $a_n = a_{n-1} + n, n > l$

$= c, n = l, c$ is a const.

Using back-substitution, yields $O(n^2)$

6. Consider the function $F(n)$ for which the pseudo code is given below:

Function $F(n)$ begin

$F1 \leftarrow 1$

If ($n = 1$) then $F \leftarrow 3$ else

For $i = 1$ to n do begin

$C \leftarrow 0$ For

$j = 1$ to $F(n - 1)$ do begin $C \leftarrow C + 1$ end

$F1 = F1 * C$ end

$F = F1$ end

[n is a positive integer greater than zero]

(a) Derive a recurrence relation for $F(n)$

(b) Solve the recurrence relation for a closed form solutions of $F(n)$. **[1992]**

Solution:

$T(n) = 3, n = 1$

$T(n) = n + T(n - 1)K$,

$n > 1$; 'k' is a constant.