	b)	Solve the following 0/1 knapsack problem using least cost branch and bound method. $n=4$, $(p1, p2, p3, p4) = (10, 10, 12, 18)$, $(w1, w2, w3, w4) = (2, 4, 6, 9)$, knapsack capacity $m=15$.	L3	CO4	5 M
OR					
11	a)	Apply Least cost branch and bound method to solve the TSP for the following cost matrix $$C=\begin{bmatrix} \infty & 4 & 8 & 3 \\ 2 & \infty & 3 & 6 \\ 5 & 8 & \infty & 2 \\ 7 & 6 & 3 & \infty \end{bmatrix}$$	L3	CO4	5 M
	b)	Explain the classes of NP hard and NP complete.	L2	CO4	5 M

Code: 23CS3301, 23AM3301, 23DS3301

PVP 23

II B.Tech - I Semester – Regular / Supplementary Examinations
NOVEMBER 2025

ADVANCED DATA STRUCTURES AND ALGORITHM ANALYSIS

(Common for CSE, AIML, DS)

Duration: 3 hours
Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

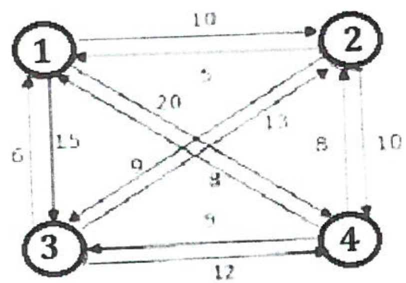
BL – Blooms Level
CO – Course Outcome

PART – A

		BL	CO
1.a)	What is meant by an algorithm?	L2	CO1
1.b)	Define Time complexity of an algorithm.	L1	CO1
1.c)	Differentiate between adjacency matrix and adjacency list.	L2	CO1
1.d)	What are disjoint sets?	L2	CO1
1.e)	What are the advantages and disadvantages of divide and conquer method?	L2	CO1
1.f)	Define minimum cost spanning tree.	L1	CO1
1.g)	What is the Time complexity of optimal binary search tree?	L2	CO1
1.h)	State one merit and one demerit of Bellman ford algorithm.	L2	CO1
1.i)	What are the general steps followed in backtracking?	L2	CO1
1.j)	Explain the classes of P and NP.	L2	CO1

PART – B

			BL	CO	Max. Marks
		UNIT-I			
2	a)	List and explain characteristics of an algorithm.	L2	CO1	5 M
	b)	Explain about the asymptotic notations with suitable examples.	L2	CO1	5 M
		OR			
3	a)	Write an algorithm to find sum of first n natural numbers.	L2	CO1	5 M
	b)	Explain non recursive algorithm for finding first n terms of Fibonacci sequence and analyze its time complexity.	L2	CO1	5 M
		UNIT-II			
4	a)	Construct min-heap for the following data 50,80,40,60,45,30,35	L3	CO3	5 M
	b)	Write an algorithm for simple union.	L3	CO3	5 M
		OR			
5	a)	Define Graph. Explain how graphs are represented.	L2	CO3	5 M
	b)	Explain Depth first Search with an example.	L3	CO3	5 M
		UNIT-III			
6		Explain in detail the average case analysis of quick sort.	L3	CO2	10 M

		OR			
7		Explain Prim's and Kruskal's algorithms with an example.	L3	CO2	10 M
		UNIT-IV			
8		Find an optimal tour for the travelling sales person problem in the following graph by using dynamic programming. 	L3	CO2	10 M
		OR			
9		Using algorithm OBST compute w(i, j), r(i, j), $0 \leq i \leq j \leq 4$ for the identifier set (a1, a2, a3, a4) = (end, goto, print, stop) with p(1) = 3, p(2) = 3 ,p(3) = 1, p(4) = 1, q(0) = 2, q(1) = 3, q(2) = 1, q(3) = 1, q(4) = 1, q(5) = 1. Using r(i, j), construct the optimal binary search tree.	L4	CO3	10 M
		UNIT-V			
10	a)	Apply Backtracking method to solve the following sum of subsets problem S={2, 3, 5, 8, 10, 15} , n=6, m=20.	L3	CO4	5 M

Schema

PVP 23

Code: 23CS3301, 23AM3301, 23DS3301

II B.Tech - I Semester – Regular / Supplementary Examinations
NOVEMBER 2025

ADVANCED DATA STRUCTURES AND ALGORITHM ANALYSIS

(Common for CSE, AIML, DS)

Duration: 3 hours   Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level   CO – Course Outcome

Part A

1.a) What is meant by an algorithm. (2M)
Definition/description - 2M

1.b) Define Time complexity of an algorithm (2M)
Definition - 2M

1.c) Differentiate between adjacency matrix and adjacency list. (2M)
Adjacency matrix-1M
Adjacency list-1M

1.d) What are disjoint sets? (2M)
Definition/explanation - 2M

1.e) What are the advantages and disadvantages of divide and conquer method? (2M)
Any one advantage- 1M
Any one disadvantage- 1M

1.f) Define minimum cost spanning tree. (2M)
Definition/explanation - 2M

1.g) What is the Time complexity of optimal binary search tree? (2M)
Time complexity – 2M

1.h) State one merit and one demerit of Bellman ford algorithm. (2M)
Any one merit – 1M
Any one demerit – 1M

1.i) What are the general steps followed in backtracking? (2M)
Steps/explanation of backtracking – 2M

1.j) Explain the classes of P and NP. (2M)
P definition– 1M
NP defintion-1M

....

Part B

Unit -1

2. a) List and explain characteristics of an algorithm (5M)
5 characteristics – 5M

2.b) Explain about the asymptotic notations with suitable examples. (5M)
5 asymptotic notations – 5M

OR

3. a) Write an algorithm to find sum of first n natural numbers. (5M)
Algorithm – 5M

....

3. b) Explain non recursive algorithm for finding first n terms of Fibonacci sequence and analyze its time complexity. (5M)
Algorithm – 3M
Analysis-2M

....

Unit -2

4.a) Construct min-heap for the following data: 50, 80, 40, 60, 45, 30, 35 (5M)
Construction – 5M

....

4.b) Write an algorithm for simple union (5M)
Algorithm – 5M

....

OR

5.a) Define Graph. Explain how graphs are represented (5M)
Graph definition – 1M
Any two representations-4M

....

5. b) Explain Depth first Search with an example. (5M)
Any example with proper explanation - 5M

....

Unit -3

6. Explain in detail the average case analysis of quick sort. (10M)

Answer – 2M
Explanation-8M

....

OR

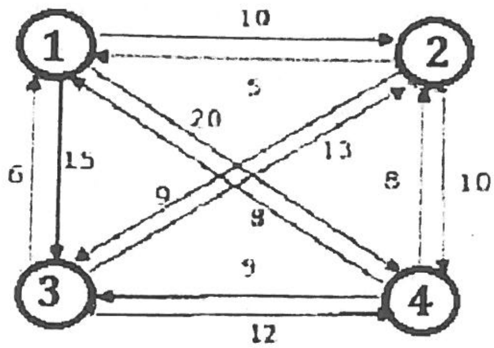
7) Explain Prim's and Kruskal's algorithms with an example. (10M)

Any example for Prim's algorithm with proper explanation -5M
Any example for Kruskal's algorithm with proper explanation -5M

....

Unit -4

8. Find an optimal tour for the travelling sales person problem in the following graph by using dynamic programming. (10M)



Calculation of values/Procedure-8M
Optimal tour with cost-2M

....

OR

9. Using algorithm OBST, computer $w(i,j), r(i,j)$ and $c(i,j)$, $0 \leq i \leq j \leq 4$ for the identifier set (a1, a2, a3, a4)=(end, goto, print, stop) with p(1)=3, p(2)=3, p(3)=1, p(4)=1, q(0)=2, q(1)=3, q(2)=1,q(3)=1,q(4)=1,q(5)=1. Using r(i, j), construct optimal binary search tree. (10 M)

Calculation of values/Procedure-8M
OBST with optimal cost-2M

....

Unit -5

10.a) Apply Backtracking method to solve the following sum of subsets problem

$$S = \{2, 3, 5, 8, 10, 15\}, n = 6, m = 20.\text{(5M)}$$

State space tree– 4M
Answer-1M

....

10.b) Solve the following 0/1 knapsack problem using least cost branch and bound method n=4, (p1, p2, p3, p4) = (10, 10, 12, 18), (w1, w2, w3, w4) = (2, 4, 6, 9), knapsack capacity m=15. (5M)

State space tree– 4M
Answer-1M

....

OR

11.a) Apply Least cost branch and bound method to solve the TSP for the following cost matrix(5M)

$$C = \begin{bmatrix} \infty & 4 & 8 & 3 \\ 2 & \infty & 3 & 6 \\ 5 & 8 & \infty & 2 \\ 7 & 6 & 3 & \infty \end{bmatrix}$$

State space tree/Calculation – 4M
Answer/Optimal tour-1M

11.b) Explain the classes of NP hard and NP complete (5M)

NP hard -2M
NP complete-2M
Relation between NP hard and NP complete-1M

....

Key

PVP 23

Code: 23CS3301, 23AM3301, 23DS3301

II B.Tech - I Semester – Regular / Supplementary Examinations
NOVEMBER 2025

ADVANCED DATA STRUCTURES AND ALGORITHM ANALYSIS

(Common for CSE, AIML, DS)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

Part A

1.a) What is meant by an algorithm. (2M)

Definition/description - 2M

An algorithm is a finite set of instructions that, if followed, accomplishes a task.

An algorithm is composed of a finite set of steps, each of which may require one or more operations

1.b) Define Time complexity of an algorithm (2M)

Definition - 2M

The amount of time that is needed to execute an algorithm is called time complexity. In general, time complexity of an algorithm is measured in terms of the number of basic operations performed by the algorithm.

1.c) Differentiate between adjacency matrix and adjacency list. (2M)

Adjacency matrix-1M
Adjacency list-1M

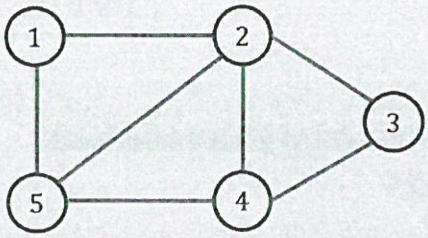
Let $G = (V, E)$ be a graph with $|V| = n$.

Let the vertices are numbered $1, 2, \ldots\ldots, n$.

The adjacency matrix represented of a graph G consists of a $n \times n$ matrix.

$$A = [a_{ij}]_{n\times n} \quad ,\text{where } a_{ij} = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & otherwise \end{cases}$$

Adjacency matrix: Example



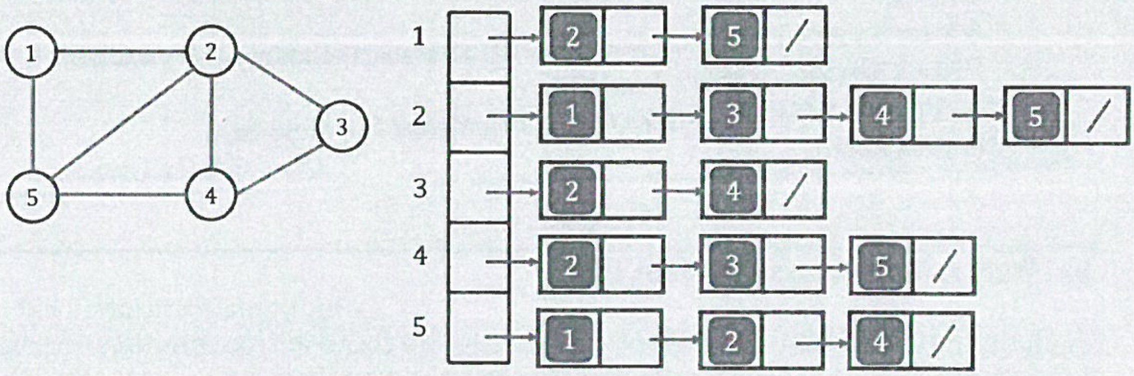
Adjacency matrix $A=$
$$\begin{array}{c} \\ 1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{ccccc}1&2&3&4&5\end{array}\\ \begin{bmatrix}0&1&0&0&1\\1&0&1&1&1\\0&1&0&1&0\\0&1&1&0&1\\1&1&0&1&0\end{bmatrix}\end{array}$$

Adjacency list representation of a graph $G = (V, E)$ consists of an array Adj of $|V|$ lists, one for each vertex in V.

For each $u \in V$, the adjacency list $\text{Adj}[u]$ contains all the vertices $v \ni$ there is an edge $(u, v) \in E$.

i.e., $\text{Adj}[u]$ consists of all the vertices adjacent to u in G.

Adjacency list: Example



1.d) What are disjoint sets? (2M)

Definition/explanation - 2M

- A Disjoint set S is a collection of sets $S_1, \ldots\ldots S_n$ where $\forall_{i \neq j} S_i \cap S_j = \phi$
- Each set has a representative which is a member of the set (Usually the minimum if the elements are comparable) .

1.e) What are the advantages and disadvantages of divide and conquer method? (2M)

Any one advantage- 1M
Any one disadvantage- 1M

Advantage (one point):

- Breaks a large problem into smaller subproblems, making it easier and faster to solve.

Disadvantage (one point):

- Uses more memory due to recursive calls and storage of intermediate results.

1.f) Define minimum cost spanning tree. (2M)

Definition/explanation - 2M

Spanning tree with the minimum sum of weights is called minimal spanning tree or minimum cost spanning tree.

1.g) What is the Time complexity of optimal binary search tree? (2M)

Time complexity – 2M

Time complexity of optimal binary search tree=$O(n^3)$

- .

1.h) State one merit and one demerit of Bellman ford algorithm. (2M)

Any one merit – 1M
Any one demerit – 1M

Merit: Works for graphs with negative weights also
Demerit: Will not work when the graph is having cycle of negative length

1.i) What are the general steps followed in backtracking? (2M)

Steps/explanation of backtracking – 2M

Steps followed in backtracking

1. Choose a possible option and proceed recursively.
2. If the chosen option leads to a dead end, backtrack by undoing the last step.
3. Continue exploring other options until a valid solution is found.

1.j) Explain the classes of P and NP. (2M)

P definition– 1M
NP defintion-1M

Class P:

Class P is the set of all decision problems solvable by deterministic algorithms in polynomial time

Class NP:

Class NP is the set of all decision problems solvable by non-deterministic algorithms in polynomial time

....

Part B

Unit -1

2. a) List and explain characteristics of an algorithm (5M)

5 characteristics – 5M

An algorithm is a finite set of instructions that, if followed, accomplishes a task. An algorithm is composed of a finite set of steps, each of which may require one or more operations

The following are the characteristics of an algorithm

1. Input: An algorithm takes zero or more inputs

2. Output: An algorithm produces at least one output or result based on the provided inputs

3. Definiteness: Each step or instruction in an algorithm must be clear, unambiguous, and precisely defined

4. Finiteness: An algorithm should terminate after a finite number of well-defined steps or instructions

5. Effectiveness: Each instruction must be feasible and very basic so that it can carried out, in principle, by a person using pencil and paper

....

2.b) Explain about the asymptotic notations with suitable examples. (5M)

5 asymptotic notations – 5M

Notations which are used to describe or represent asymptotic complexity of an algorithm are called asymptotic notations.

They describe the running time/ space requirement of an algorithm for large values of n ($n \to \infty$).

In asymptotic notations, we abstract away low-order terms and constant factors.

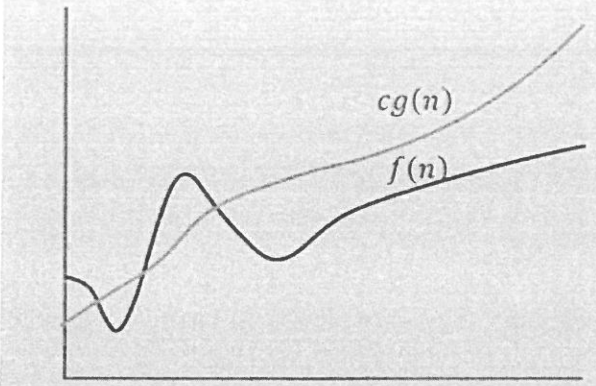
There are 5 types of asymptotic notations. They are

1. Big-oh Notation (O)
2. Big-omega Notation (Ω)
3. Theta Notation (Θ)
4. Little-oh Notation (o)
5. Little-omega Notation (ω)

1. Big-oh Notation (O):

$f(n)$ *is* $O(g(n))$ means there exist positive constants c and n_0, such that

$$f(n) \leq cg(n) \quad \forall\ n \geq n_0$$



$g(n)$ has larger or equal order of growth as $f(n)$

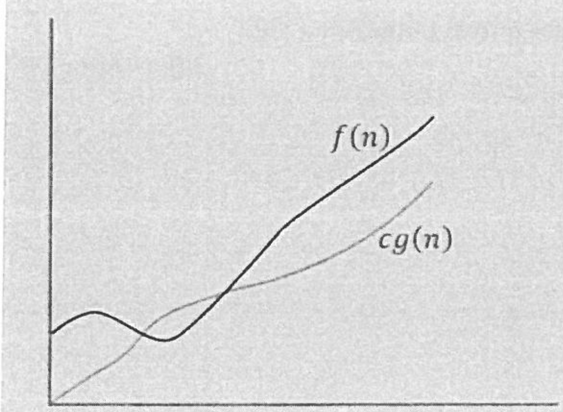
$g(n)$ is an asymptotic upper bound for $f(n)$

Example: $2n^2 + 5n = O(n^2)$

2. Big-omega Notation (Ω):

$f(n)$ *is* $\Omega\left(g(n)\right)$ means there exist positive constants c and n_0, such that

$$0 \leq cg(n) \leq f(n)\, \forall\, n \geq n_0$$



$g(n)$ has smaller or equal order of growth as $f(n)$

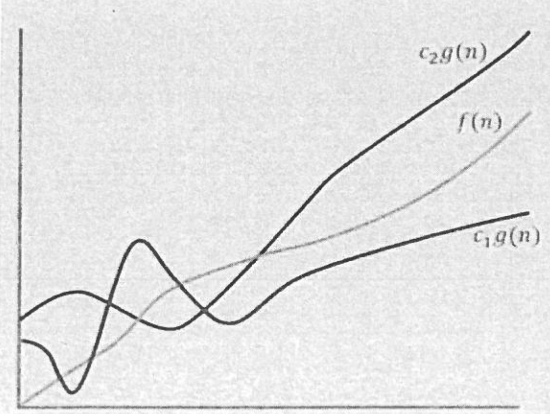
$g(n)$ is an asymptotic lower bound for $f(n)$

Example: $2n^2 + 5n = \Omega(n^2)$

3. Theta Notation (Θ):

$f(n)$ *is* $\Theta(g(n))$ means there exist positive constants c_1, c_2 and n_0, such that

$$\leq c_1\, g(n) \leq f(n) \leq c_2\, g(n)\, \forall\, n \geq n_0$$



$f(n)$ grows with same order as $g(n)$.

$g(n)$ is an asymptotically tight bound for $f(n)$

Example: $2n^2 + 5n = \Theta(n^2)$

4. Little-oh Notation (o):

$f(n)$ is $o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

Example: $2n^2 + 5n = o(n^3)$

5. Little-omega Notation (ω):

$f(n)$ is $\omega(g(n))$ if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

Example: $2n^2 + 5n = \omega(n)$

....

OR

3. a) Write an algorithm to find sum of first n natural numbers. (5M)

Algorithm – 5M

```
Algorithm Fun(n)
{
    sum:=0;
    for i:= 1 to n do
        sum:=sum+i;
    return(sum);
}
```

....

3. b) Explain non recursive algorithm for finding first n terms of Fibonacci sequence and analyze its time complexity. (5M)

Algorithm – 3M
Analysis-2M

Algorithm/Pseudo code	s/e	freq	total
`Algorithm Fibonacci(n)`	0	–	0
`{`	0	–	0
`if n ≥ 1, print 0`	2	1	2
`if n ≥ 2, print 1`	2	1	2
`a:= 0,  b:=1`	2	1	2
`for i ← 3 to n do`	2	$n-2$	$2n-4$
`{`	0	–	0
`c := a + b`	2	$n-2$	$2n-4$
`print c`	1	$n-2$	$n-2$
`a := b`	1	$n-2$	$n-2$
`b := c`	1	$n-2$	$n-2$
`}`	0	–	0
`}`	0	–	0

Time Complexity $= 7n - 8$

$= \Theta(n)$

Note: Any analysis, which will number of operations is proportional to $O(n)$

....

Unit -2

4.a) Construct min-heap for the following data: 50, 80, 40, 60, 45, 30, 35 (5M)

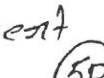
Construction – 5M

We can construct min-heap by using any one of the following 2 methods

Method 1: Construction by inserting elements one by one and adjusting heap property

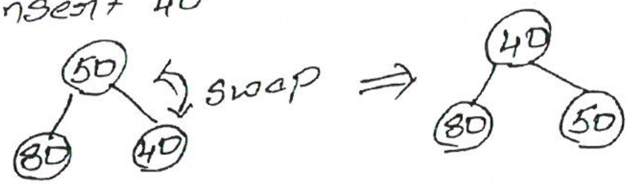
Data: 50, 80, 40, 60, 45, 30, 35

1. Insert 50

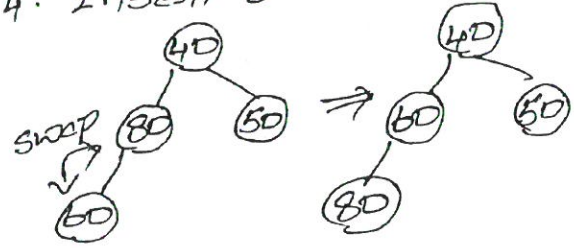


2. Insert 80

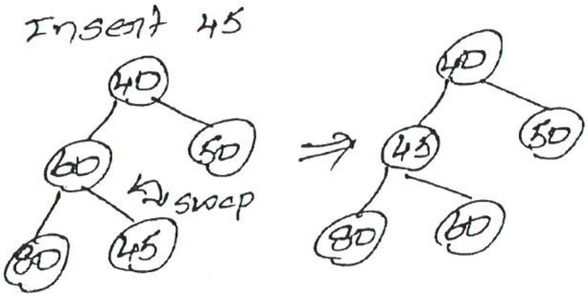
3. Insert 40



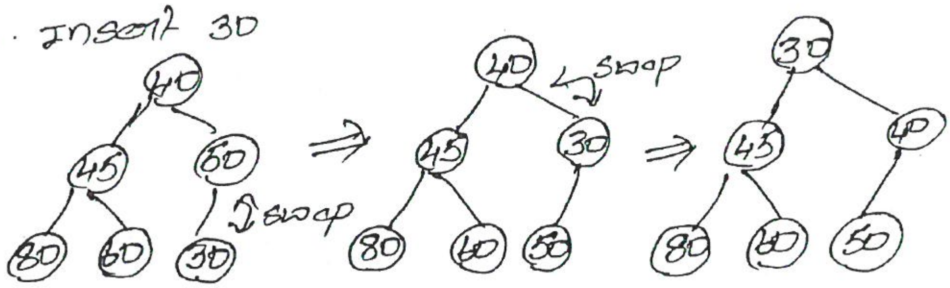
4. Insert 60



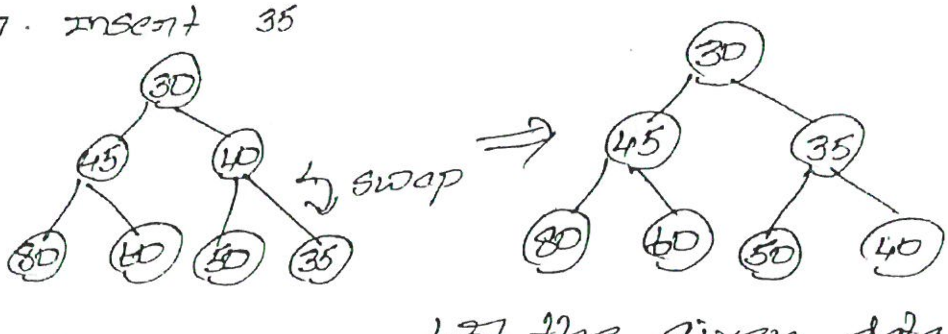
5 Insert 45



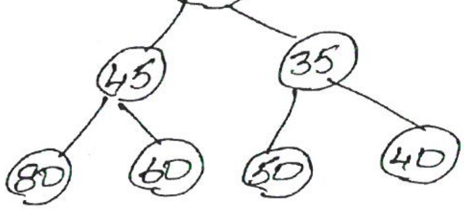
6. Insert 30



7. Insert 35

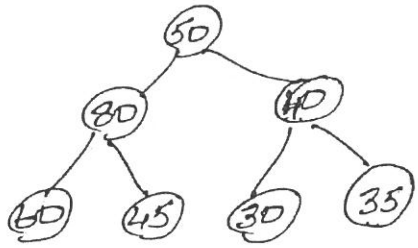


∴ Min-heap for the given data is

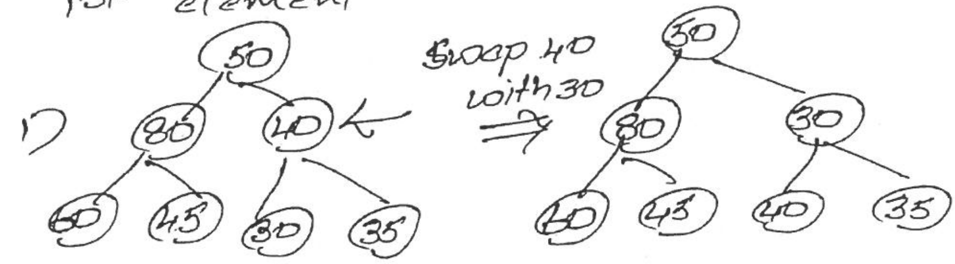


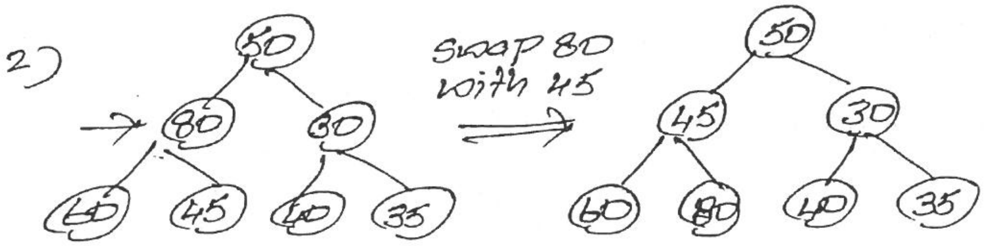
Method 2: Construction by inserting elements one by one and adjusting heap property

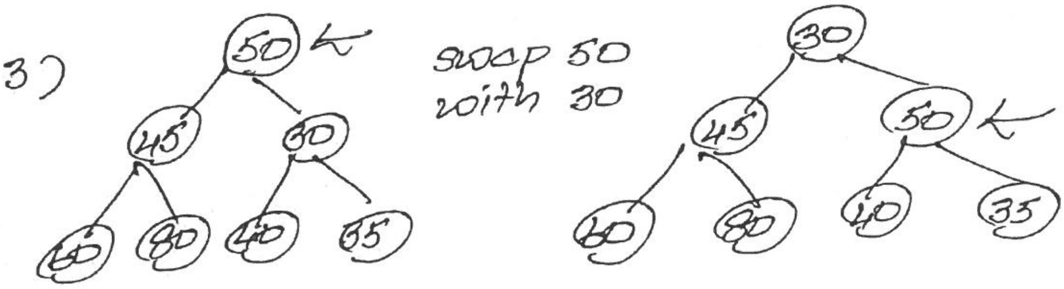
First, we can insert all keys with order property. Then we start heapify process from $(\frac{n}{2})^{th}$ node to 1st node

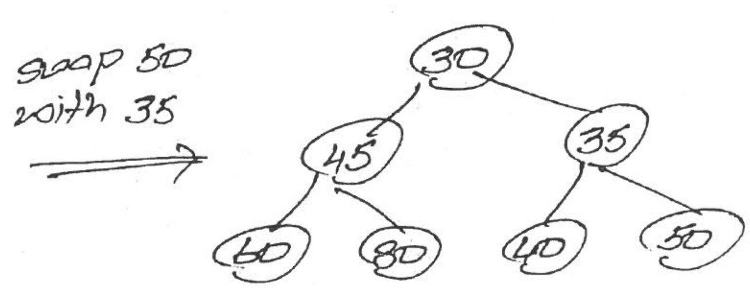


we do MinHeapify from $\frac{n}{2}^{th}$ element to 1st element

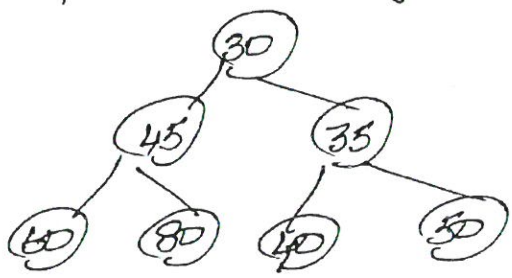








∴ Min heap for the given data is



....

4.b) Write an algorithm for simple union (5M)

Algorithm – 5M

```
//Returns root of the tree which contains x
Algorithm Union(x, y)
{
    u=Find(x); //Finding root of tree which contains x
    v=Find(y); //Finding root of tree which contains y
    if(u!=v)
        parent[v]=u;//Attaching tree v to tree u
}
```

....

OR

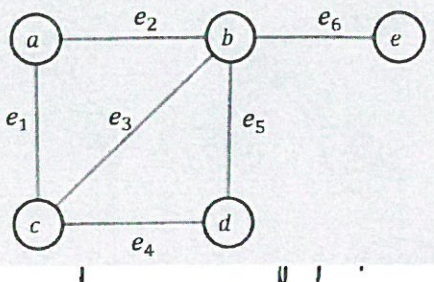
5.a) Define Graph. Explain how graphs are represented (5M)

Graph definition – 1M

Any two representations-4M

A graph is a pair (V, E) of sets. The elements of V are called vertices and the elements of E are called edges.

Example:



$V = \{a, b, c, d, e\}$

$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

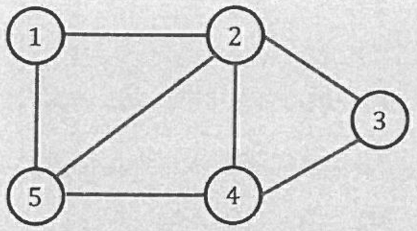
$E = \{\{a, c\}, \{a, b\}. \{b, c\}, \{c, d\}, \{b, d\}, \{b, e\}\}$

- There are 3 standard ways to represent a graph $G = (V, E)$
  1. Adjacency matrix
  2. Adjacency list
  3. Incidence matrix

Adjacency matrix:-

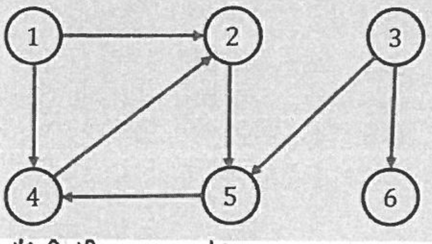
- Let $G = (V, E)$ be a graph with $|V| = n$.
- Let the vertices are numbered $1, 2, \ldots, n$
- The adjacency matrix represented of a graph G consists of a $n \times n$ matrix.

$$A = [a_{ij}]_{n \times n} \quad , \text{where} \quad a_{ij} = \begin{cases} 1 & \text{if } [i,j] \in E \\ 0 & \text{otherwise} \end{cases}$$



Adjacency matrix $A=$

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 1 & 1 & 1 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 & 0 & 1 \\ 5 & 1 & 1 & 0 & 1 & 0 \end{array}$$



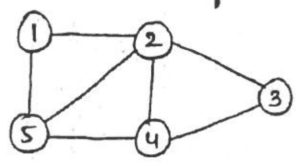
Adjacency matrix $A=$

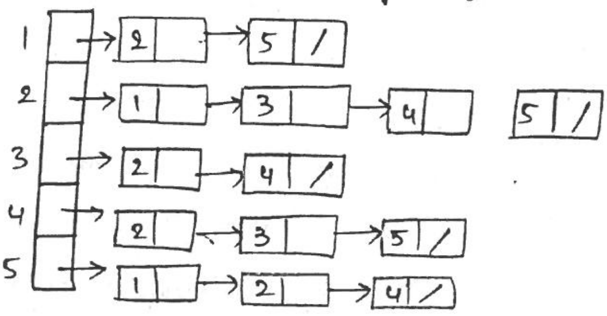
$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 1 \\ 4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 1 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

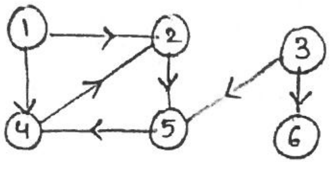
* Adjacency List:-

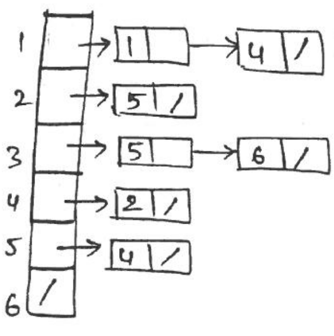
- In this representation we create an array of size n where each element of array is a linked list of adjacency vertices

Eg:-





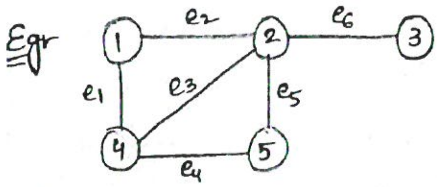




* **Incidence matrix:**

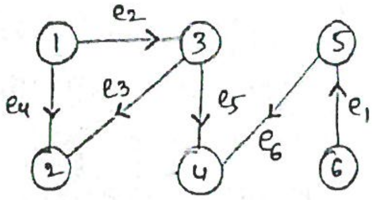
- An incidence matrix is a matrix where rows correspond to vertice and columns correspond to edges.
- Each element in the matrix indicates whether a particular vertex is incident to a particular edge.

Eg:



Incidence matrix $I =$

	e_1	e_2	e_3	e_4	e_5	e_6
1	1	1	0	0	0	0
2	0	1	1	0	1	1
3	0	0	0	0	0	1
4	1	0	1	1	0	0
5	0	0	0	1	1	0



$I =$

	e_1	e_2	e_3	e_4	e_5	e_6
1	0	−1	0	−1	0	0
2	0	0	+1	+1	0	0
3	0	+1	−1	0	−1	0
4	0	0	0	0	+1	+1
5	+1	0	0	0	0	−1
6	0	0	0	0	0	0

....

5. b) Explain Depth first Search with an example. (5M)

Any example with proper explanation - 5M

The strategy followed by Depth - First Search is to search deeper in the graph whenever possible.

Main idea:

Mark s as visited. Push s on to the stack.

Repeat the following till the stack is empty:

Pop a vertex (say w) from the stack.

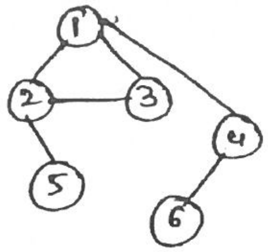
For each neighbour x of w that is not yet visited

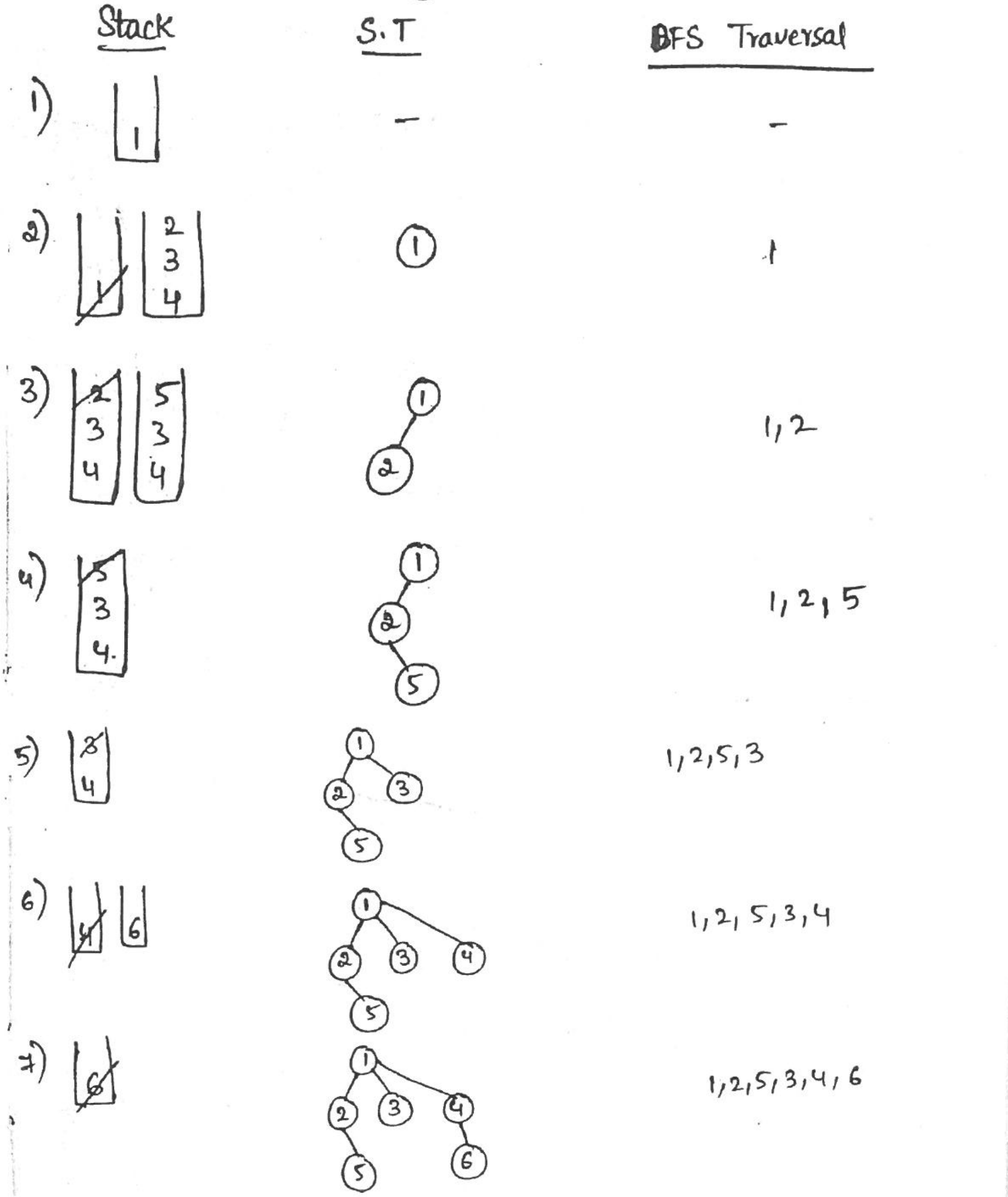
Mark x as visited.

Add x to the stack.

Mark w as processed.

Example: Consider the following graph





Note: We can follow any strategy which the follows the DFS principle " Search the graph as deeper as possible whenever possible"

....

Unit -3

6. Explain in detail the average case analysis of quick sort. (10M)

Answer – 2M
Explanation-8M

The RR for time complexity of quicksort is given by

$$T(n) = 0 \quad \text{if } n < 2$$
$$= T(k) + T(n-k-1) + n \quad \text{if } n \geq 2 \quad \rightarrow (1)$$

The above RR can be used to find the time complexity of the quicksort in three situations: worst case, best case, average case

Average case

$$T(n) = \frac{1}{n} \sum_{k=0}^{n-1} [T(k) + T(n-k-1)] + n$$

$$= \frac{1}{n} [T(0) + T(n-1) + T(1) + T(n-2) + T(2) + T(n-3) + \cdots + T(n-2) + T(1) + T(n-1) + T(0)] + n$$

$$= \frac{1}{n} 2 [T(0) + T(1) + T(2) + \cdots + T(n-1)] + n$$

$$\therefore n T(n) = 2[T(0) + T(1) + T(2) + \cdots + T(n-1)] + n^2 \rightarrow (2)$$

Replace 'n' with 'n-1' in (2), then we have

$$(n-1) T(n-1) = 2[T(0) + T(1) + \cdots + T(n-2)] + (n-1)^2 \rightarrow (3)$$

From (2) × (3), we get

$$n T(n) - (n-1) T(n-1) = 2 T(n-1) + n^2 - (n-1)^2$$

$$\Rightarrow n T(n) = (n+1) T(n-1) + 2n - 1$$

$$\Rightarrow \frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2n-1}{n(n+1)}$$

$$\Rightarrow \frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2}{n+1} - \frac{1}{n(n+1)}$$

$$\Rightarrow \frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2}{n+1} - \left[\frac{1}{n} - \frac{1}{n+1}\right]$$

$$\Rightarrow \frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{3}{n+1} - \frac{1}{n} \rightarrow (4)$$

Assume $b_n = \frac{T(n)}{n+1}$ Then ④ becomes

$$b_n = b_{n-1} + \frac{3}{n+1} - \frac{1}{n}$$

$$= \left[b_{n-2} + \frac{3}{n} - \frac{1}{n-1}\right] + \left[\frac{3}{n+1} - \frac{1}{n}\right]$$

$$= b_{n-2} + \left[\frac{3}{n} - \frac{1}{n-1} + \frac{3}{n+1} - \frac{1}{n}\right]$$

$$= \left[b_{n-3} + \frac{3}{n-1} - \frac{1}{n-2}\right] + \left[\frac{3}{n} - \frac{1}{n-1} + \frac{3}{n+1} - \frac{1}{n}\right]$$

$$\vdots$$

$$= b_{n-n} + \left[\left(\frac{3}{2} - \frac{1}{1}\right) + \left(\frac{3}{3} - \frac{1}{2}\right) + \cdots + \frac{3}{n} - \frac{1}{n-1} + \frac{3}{n+1} - \frac{1}{n}\right]$$

$$= b_0 + \left[\frac{3}{n+1} + \frac{3}{n} + \frac{3}{n-1} + \cdots + \frac{3}{2}\right] - \left[\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{2} + 1\right]$$

$$= b_0 + \frac{3}{n+1} + \left[\frac{2}{n} + \frac{2}{n-1} + \cdots + \frac{2}{2}\right] - 1$$

$$= b_0 - 1 + \frac{3}{n+1} + 2\left[\frac{1}{n} + \frac{1}{n-1} + \cdots \frac{1}{2} + 1\right] - 2$$

$$= \frac{T(0)}{0+1} - 3 + \frac{3}{n+1} + 2\sum_{K=1}^{n} \frac{1}{K}$$

$$\simeq 2\sum_{K=1}^{n} \frac{1}{K} = 2\int_1^n \frac{1}{K} = 2\left[\log K\right]_1^n = 2\left[\log n - \log 1\right]$$

$$= 2\log n$$

$$\therefore b_n \simeq 2\log n$$

$$\Rightarrow \frac{T(n)}{n+1} \simeq 2(n+1)\log n$$

$$\therefore T(n) = \Theta(n\log n)$$

....

OR

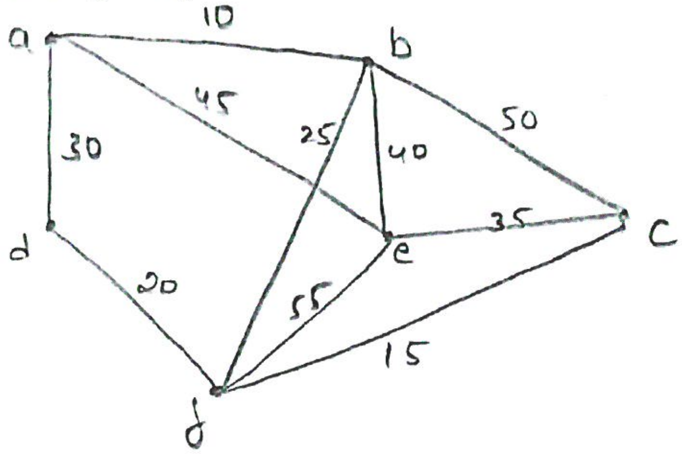
7) Explain Prim's and Kruskal's algorithms with an example. (10M)

Any example for Prim's algorithm with proper explanation -5M

Any example for Kruskal's algorithm with proper explanation -5M

Prim's algorithm:

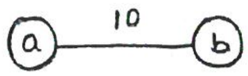
Consider the following example



1. Start with vertex 'a'

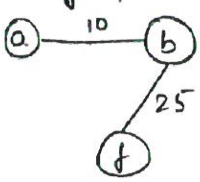
2. Fringe edges : ab, ad, ae

Add ab to MST



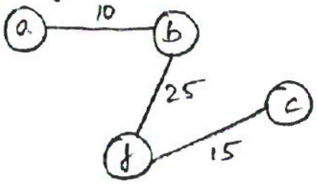
3. Fringe edges : ad, ae, bc, bf, be

Add bf to MST



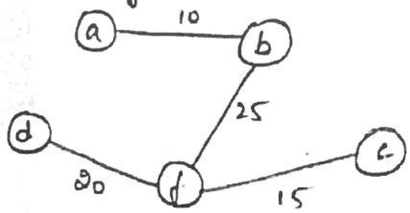
4. Fringe edges: ad, ae, bc, be, fd, fc, fe

Add fc to MST



5. Fringe edges: ad, ae, bc, be, fd, fe, ce

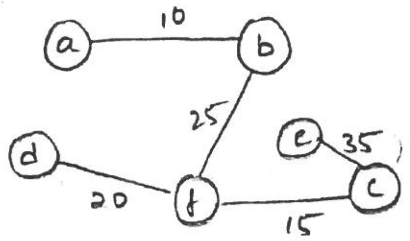
Add fd to MST



6. Fringe edges: ad, ae, bc, be, fe, ce

ad forms a cycle

add ce to MST



∴ The cost of the minimal spanning tree T

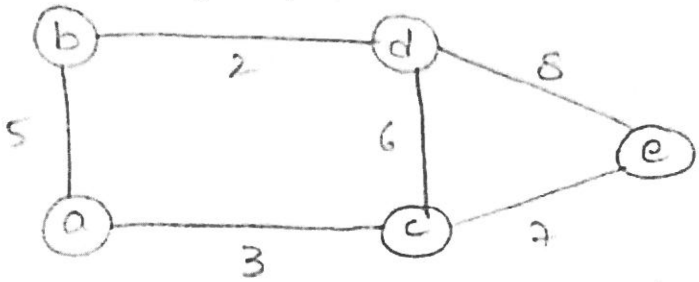
$= 10 + 25 + 20 + 15 + 35$

$= 105$

Note: Any procedure of Prim's algorithm can be explained

Kruskal's algorithm:

Consider the following example

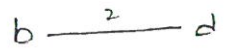


Arrange the edges in non-decreasing order of their weights

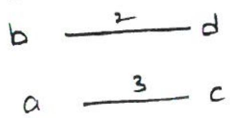
edge	bd	ac	ab	cd	ce	de
weight	2	3	5	6	7	8
status	✓	✓	✓	✗	✓	✗

1. Initially spanning tree T= ∅

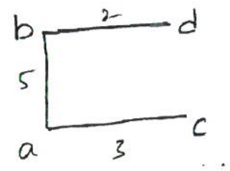
2. The edge 'bd' is having minimum weight. So, add the edge 'bd' to T



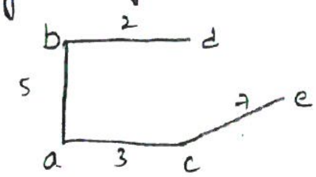
3. Add the edge ac to T



4. ab is the edge with next minimal weight add 'ab' to T



5. The next minimal weight edge is cd but the addition of it to T forms a cycle. So don't add it. The next minimum weight edge is ce. So, add it to T



we got n-1 edges

∴ The cost of the minimal Spanning tree T

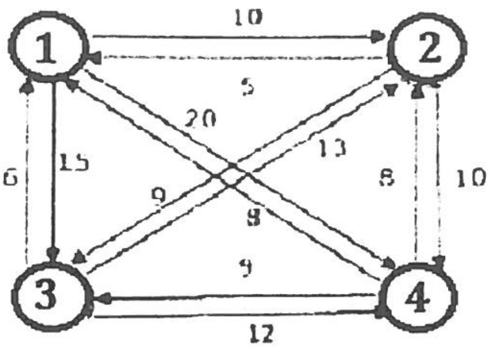
$= 2+3+5+7=17$

Note: Any procedure of kruskal algorithm can be explained

....

Unit -4

8. Find an optimal tour for the travelling sales person problem in the following graph by using dynamic programming. (10M)



Calculation of values/Procedure-8M
Optimal tour with cost-2M

The cost matrix of the given graph is

$$\text{cost matrix } C = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

Let $g(i, S)$ be the length of a shortest path starting at vertex i, going through all vertieces of S.

we know that $g(i, S) = \min_{j \in S} \{ c_{ij} + g(j, S - \{j\}) \}$ → ①

and $g(i, \phi) = c_{i1}, \quad 1 \le i \le n$

$g(2, \phi) = c_{21} = 5$

$g(3, \phi) = c_{31} = 6$

$g(4, \phi) = c_{41} = 8$

Now by using ①, we have

$g(2, \{3\}) = c_{23} + g(3, \phi) = 9 + 6 = 15$

$g(2, \{4\}) = c_{24} + g(4, \phi) = 10 + 8 = 18$

$g(3, \{2\}) = c_{32} + g(2, \phi) = 13 + 5 = 18$

$g(3, \{4\}) = c_{34} + g(4, \phi) = 12 + 8 = 20$

$g(4, \{2\}) = c_{42} + g(2, \phi) = 8 + 5 = 13$

$g(4, \{3\}) = c_{43} + g(3, \phi) = 9 + 6 = 15$

Now let us compute $g(i, S)$ with $|S| = 2, i \notin S, i \neq 1, 1 \notin S$

$g(2, \{3, 4\}) = \min\{c_{23} + g(3, \{4\}), c_{24} + g(4, \{3\})\}$

$= \min\{9 + 20, 10 + 15\}$

$= \min\{29, 25\} = 25$

$g(3, \{2, 4\}) = \min\{c_{32} + g(2, \{4\}), c_{34} + g(4, \{2\})\}$

$= \min\{13 + 18, 12 + 13\}$

$= \min\{31, 25\} = 25$

$g(4, \{2, 3\}) = \min\{c_{42} + g(2, \{3\}), c_{43} + g(3, \{2\})\}$

$= \min\{8 + 15, 9 + 18\}$

$= \min\{23, 27\} = 23$

Finally, from ①, we have

$g(1, \{2, 3, 4\}) = \min\{c_{12} + g(2, \{3, 4\}), c_{13} + g(3, \{2, 4\}), c_{14} + g(4, \{2, 3\})\}$

$= \min\{10 + 25, 15 + 25, 20 + 23\}$

$= \min\{35, 40, 43\}$

$= 35$

An optimal tour of the given graph has length 35. A tour of this length can be constructed if we retain with each $g(i, S)$ the value of j that minimizes the right-hand side of equation 1.

Let $J(i, S)$ be this value. Then $J(1, \{2, 3, 4\}) = 2$. Thus the tour starts from 1 and goes to 2. The remaining tour can be obtained from $g(2, \{3, 4\})$. So $J(2, \{3, 4\}) = 4$. This the next edge is (2,4). The remaining tour is for $g(4, \{3\})$. So $J(4, \{3\}) = 3$.

∴ the optimal tour is $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$

…………………………………………………………

OR

9. Using algorithm OBST, computer $w(i,j), r(i,j)$ and $c(i,j)$, $0 \le i \le j \le 4$ for the identifier set (a1, a2, a3, a4)=(end, goto, print, stop) with p(1)=3, p(2)=3, p(3)=1, p(4)=1, q(0)=2, q(1)=3, q(2)=1,q(3)=1,q(4)=1,q(5)=1. Using r(i, j), construct optimal binary search tree. (10 M)

Calculation of values/Procedure-8M

OBST with optimal cost-2M

Let t_{ij} be the BST containing $a_{i+1}, a_{i+2} \ldots a_j$ and $E_i, E_{i+1} \ldots E_j$

$w(i,j)$ - weight of the tree t_{ij}

= sum of the probabilities of the nodes in t_{ij}

$\therefore$ $w(i,j) = p(j) + q(j) + w(i,j-1)$ if $i<j$

$= q(i)$ if $i=j$ → ①

Let $c(i,j)$ - cost of BST t_{ij}. Then

$c(i,j) = \min_{i<k\le j} \{c(i,k-1) + c(k,j)\} + w(i,j)$ → ②

and $c(i,i) = 0$ $\forall i$, $0 \le i \le n$

Let $r(i,j)$ - be the k value for which a_k is the root of t_{ij}

- the value of k that minimizes ②

and $r(i,i) = 0$ $0 \le i \le n$

Given values are $p(1) = 3, p(2) = 3, p(3) = 1, p(4) = 1, q(0) = 2, q(1) = 3, q(2) = 1, q(3) = 1, q(4) = 1$.

It is clear that $w(i,i) = q(i), c(i,i) = 0$ and $r(i,i) = 0, 0 \le i \le 4$.

By using above equations, we can do calculations as follows

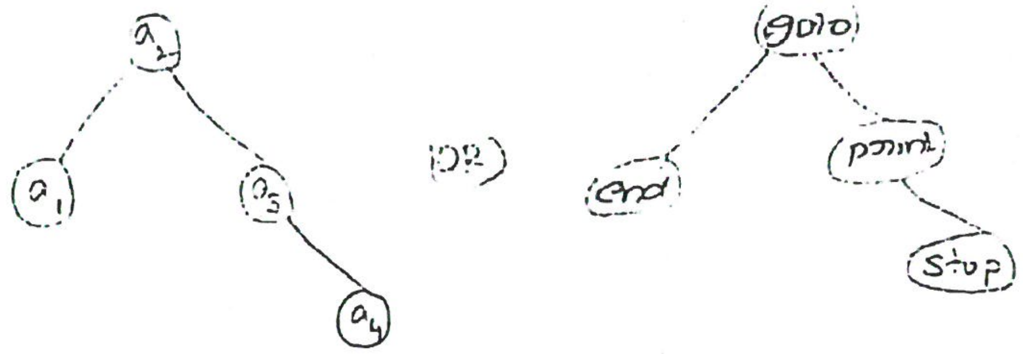
$$
\begin{aligned}
w(0,1) &= p(1) + q(1) + w(0,0) = 8 \\
c(0,1) &= w(0,1) + \min\{c(0,0) + c(1,1)\} = 8 \\
r(0,1) &= 1 \\
w(1,2) &= p(2) + q(2) + w(1,1) = 7 \\
c(1,2) &= w(1,2) + \min\ \{c(1,1) + c(2,2)\} = 7 \\
r(0,2) &= 2 \\
w(2,3) &= p(3) + q(3) + w(2,2) = 3 \\
c(2,3) &= w(2,3) + \min\ \{c(2,2) + c(3,3)\} = 3 \\
r(2,3) &= 3 \\
w(3,4) &= p(4) + q(4) + w(3,3) = 3 \\
c(3,4) &= w(3,4) + \min\ \{c(3,3) + c(4,4)\} = 3 \\
r(3,4) &= 4
\end{aligned}
$$

Knowing $w(i, i+1)$ and $c(i, i+1), 0 \leq i \leq 40$, we can again use above equations to compute $w(i, i+2), c(i, i+2)$, and $r(i, i+2), 0 < i < 3$.This process can be repeated until $w(0,4)c$, (0,4) and $r(0,4)$are obtained. The following table shows the results of this computation.

	0	1	2	3	4
0	$w_{00} = 2$ $c_{00} = 0$ $r_{00} = 0$	$w_{11} = 3$ $c_{11} = 0$ $r_{11} = 0$	$w_{22} = 1$ $c_{22} = 0$ $r_{22} = 0$	$w_{33} = 1$ $c_{33} = 0$ $r_{33} = 0$	$w_{44} = 1$ $c_{44} = 0$ $r_{44} = 0$
1	$w_{01} = 8$ $c_{01} = 8$ $r_{01} = 1$	$w_{12} = 7$ $c_{12} = 7$ $r_{12} = 2$	$w_{23} = 3$ $c_{23} = 3$ $r_{23} = 3$	$w_{34} = 3$ $c_{34} = 3$ $r_{34} = 4$	
2	$w_{02} = 12$ $c_{02} = 19$ $r_{02} = 1$	$w_{13} = 9$ $c_{13} = 12$ $r_{13} = 2$	$w_{24} = 5$ $c_{24} = 8$ $r_{24} = 3$		
3	$w_{03} = 14$ $c_{03} = 25$ $r_{03} = 2$	$w_{14} = 11$ $c_{14} = 19$ $r_{14} = 2$			
4	$w_{04} = 16$ $c_{04} = 32$ $r_{04} = 2$				

The computation is carried out by row from row 0 to row 4. From the table we see that $c(0,4) = 32$ is the minimum cost of a binary search tree for (a_1, a_2, a_3, a_4). The root of tree t_{04} is a_2. Hence, the left subtree is t_{01} and the right subtree t_{24}. Tree t_{01} has root a_1 and subtrees t_{00} and t_{11}. Tree t_{24} has root a_3; its left subtree is t_{22} and its right subtree t_{34}. Thus, with the data in the table it is possible to reconstruct t_{04}.

The optimal binary search tree is as shown below



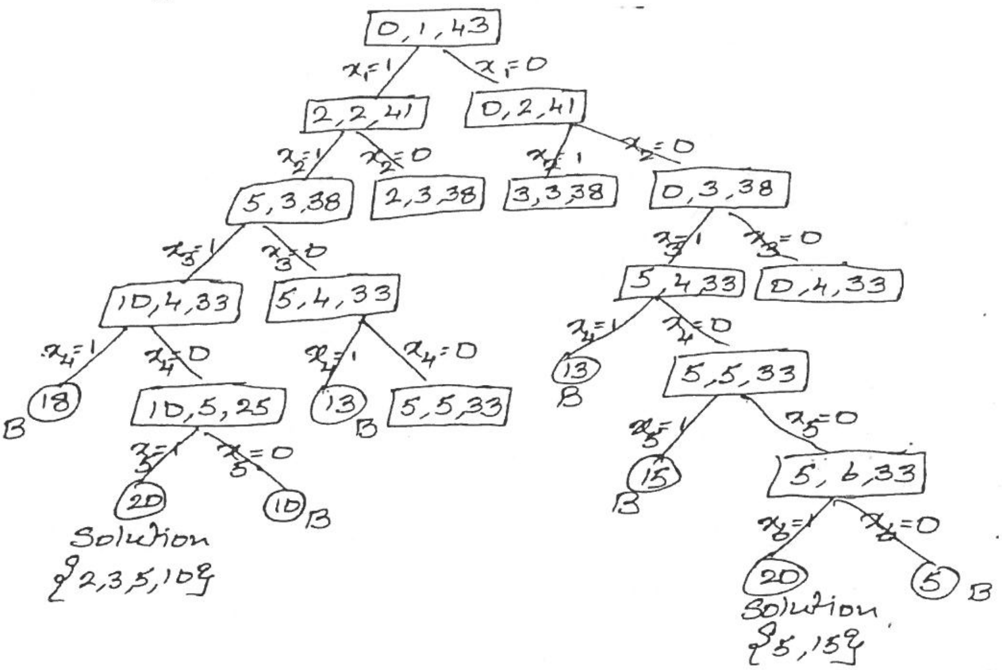
Unit -5

10.a) Apply Backtracking method to solve the following sum of subsets problem

$$S = \{2, 3, 5, 8, 10, 15\}, n = 6, m = 20. (5M)$$

State space tree– 4M
Answer-1M

Given set is S = {2,3,5,8,10,15} m = 20
The portion of the state space tree for the given problem is as follows



Solution subsets are {2,3,5,10}, {2,8,10} {2,3,15} {5,15}

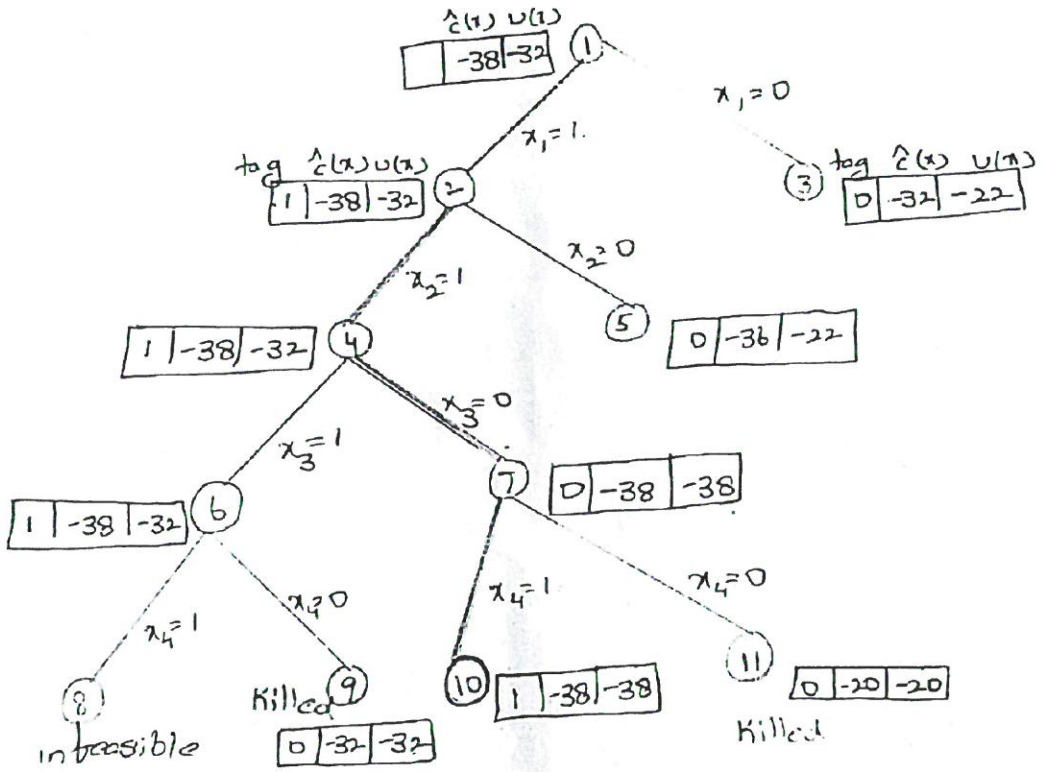
....

10.b) **Solve the following 0/1 knapsack problem using least cost branch and bound method n=4, (p1, p2, p3, p4) = (10, 10, 12, 18), (w1, w2, w3, w4) = (2, 4, 6, 9), knapsack capacity m=15. (5M)**

State space tree– 4M
Answer-1M

Given values are

(p1, p2, p3, p4) = (10, 10, 12, 18), (w1, w2, w3, w4) = (2, 4, 6, 9), m=15



Node 1:- $\hat{c}(1) = 10 + 10 + 12 + \frac{3}{9}(18) = 38$

$\hat{c}(1) = -38$

$U(1) = 10 + 10 + 12 = 32$

$U(1) = -32$

Node 2:- $\hat{c}(2) = 10 + 10 + 12 + \frac{3}{9}(18) = 38$ $\quad \hat{c}(2) = -38$

$U(2) = 10 + 10 + 12 = 32$

$U(2) = -32$

Node 3:- $\hat{c}(3) = 10 + 12 + \frac{5}{9}(18) = 32$ $\boxed{\hat{c}(3) = -32}$

$U(3) = 10 + 12 = 22$

$\boxed{U(3) = -22}$

Among nodes 2, 3 node ② is having least cost. So expand it

node 4:- $\hat{c}(4) = 10+10+12+\frac{3}{9}(18) = 38 \Rightarrow \boxed{\hat{c}(4) = -38}$

$U(4) = 10+10+12 = 32 \Rightarrow \boxed{U(4) = -32}$

node 5:- $\hat{c}(5) = 10+12+\frac{7}{9}(18) = 36 \Rightarrow \boxed{\hat{c}(5) = -36}$

$U(5) = 10+12 = 22 \Rightarrow \boxed{U(5) = -22}$

Among 3, 4, 5 node ④ is having least cost. So expand

node 6:- $\hat{c}(6) = 10+10+12+\frac{3}{9}(18) = 38 \Rightarrow \boxed{\hat{c}(6) = -38}$

$U(6) = 10+10+12 = 32 \Rightarrow \boxed{U(6) = -32}$

node 7:- $\hat{c}(7) = 10+10+18 = 38 \Rightarrow \boxed{\hat{c}(7) = -38}$

$U(7) = 10+10+18 = 38 \Rightarrow \boxed{U(7) = -38}$

Among nodes 3, 5, 6, 7 node 6, 7 is having least cost so expand it.

node 9:- $\hat{c}(9) = 10+10+12 = 32 \Rightarrow \boxed{\hat{c}(9) = -32}$

$U(9) = 10+10+12 = 32 \Rightarrow \boxed{U(9) = -32}$

node 10:- $\hat{c}(10) = 10+10+18 = 38 \Rightarrow \boxed{\hat{c}(10) = -38}$

$U(10) = 10+10+18 = 38 \Rightarrow \boxed{U(10) = -38}$

node 11:- $\hat{c}(11) = 10+10 = 20 \Rightarrow \boxed{\hat{c}(11) = -20}$

$U(11) = 10+10 = 20 \Rightarrow \boxed{U(11) = -20}$

This is solution node $(x_1, x_2, x_3, x_4) = (1,1,0,1)$.

The tag sequence for the path ⑩ ⑦ ④ ② ① is 1 0 1 1 to $x_4=1$, $x_3=0$, $x_2=1$, $x_1=1$ and optimal profit is 38.

....

OR

11.a) Apply Least cost branch and bound method to solve the TSP for the following cost matrix(5M)

$$C=\begin{bmatrix} \infty & 4 & 8 & 3 \\ 2 & \infty & 3 & 6 \\ 5 & 8 & \infty & 2 \\ 7 & 6 & 3 & \infty \end{bmatrix}$$

State space tree/Calculation – 4M

Answer/Optimal tour-1M

Given cost matrix is

$$A = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \infty & 4 & 8 & 3 \\ 2 & 2 & \infty & 3 & 6 \\ 3 & 5 & 8 & \infty & 2 \\ 4 & 7 & 6 & 3 & \infty \end{array}$$

Substract 3 2 2 3 from rows 1, 2, 3, 4

$$A = \begin{bmatrix} \infty & 1 & 5 & 0 \\ 0 & \infty & 1 & 4 \\ 3 & 6 & \infty & 0 \\ 4 & 3 & 0 & \infty \end{bmatrix} \begin{matrix} 3 \\ 2 \\ 2 \\ 3 \end{matrix}$$

(10)

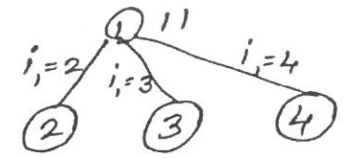
Substract 1 from column 2

$$A_1 = \begin{bmatrix} \infty & 0 & 5 & 0 \\ 0 & \infty & 1 & 4 \\ 3 & 5 & \infty & 0 \\ 4 & 2 & 0 & \infty \end{bmatrix}$$

(1)

∴ So, the cost of root node = 10+1 = 11

portion of statespace tree is as shown below



Path 1-2:

Set 1^{st} row, 2^{nd} col of A_1 to ∞

Set $A_1[2,1] = \infty$

Then the resultant matrix is $A_2 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 1 & 4 \\ 3 & \infty & \infty & 0 \\ 4 & \infty & 0 & \infty \end{bmatrix}$

~~Reduce the column 1 by 3~~

Reduce the row 2 by 1

$$A_2 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 1 & 4 \\ 3 & \infty & \infty & 0 \\ 4 & \infty & 0 & \infty \end{bmatrix} \begin{matrix} \\ -1 \\ \\ \\ \end{matrix} \Longrightarrow A_2 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 3 \\ 3 & \infty & \infty & 0 \\ 4 & \infty & 0 & \infty \end{bmatrix}$$

Reduce the column 1 by 3

$$A_2 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 3 \\ 3 & \infty & \infty & 0 \\ 4 & \infty & 0 & \infty \end{bmatrix} \Longrightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 3 \\ 0 & \infty & \infty & 0 \\ 1 & \infty & 0 & \infty \end{bmatrix}$$

-3

cost of node 2 is $\hat{c}(2) = \hat{c}(1) + A_1[1,2] + r$

$= 11 + 0 + 4 = 15$

Path 1-3:

Set 1^{st} row, 3^{rd} col of A_1 to ∞

Set $A_1[3,1] = \infty$

then the resultant matrix is $A_3 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 4 \\ \infty & 5 & \infty & 0 \\ 4 & 2 & \infty & \infty \end{bmatrix}$

Reduce the row 4 by 2

$$A_3 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 4 \\ \infty & 5 & \infty & 0 \\ 4 & 2 & \infty & \infty \end{bmatrix} \begin{matrix} \\ \\ \\ -2 \end{matrix} \Longrightarrow A_3 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 4 \\ \infty & 5 & \infty & 0 \\ 2 & 0 & \infty & \infty \end{bmatrix}$$

cost of node 3 is $\hat{c}(3) = \hat{c}(1) + A_1[1,3] + r$

$= 11 + 5 + 2 = 17$

Path 1-4:

Set 1^{st} row, 4^{th} col of A_1 to ∞

Set $A_1[4,1] = \infty$

then the resultant matrix is $A_4 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 1 & \infty \\ 3 & 5 & \infty & \infty \\ \infty & 2 & 0 & \infty \end{bmatrix}$

Reduce row 3 by 3

$$A_4 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 1 & \infty \\ 3 & 5 & \infty & \infty \\ \infty & 2 & 0 & \infty \end{bmatrix} -3 \Rightarrow A_4 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 1 & \infty \\ 0 & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty \end{bmatrix}$$

Reduce the colum 2 by 2

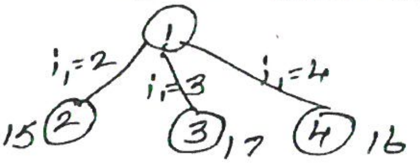
$$A_4 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 1 & \infty \\ 0 & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty \end{bmatrix} \Rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 1 & \infty \\ 0 & 0 & \infty & \infty \\ \infty & 0 & 0 & \infty \end{bmatrix}$$

-2

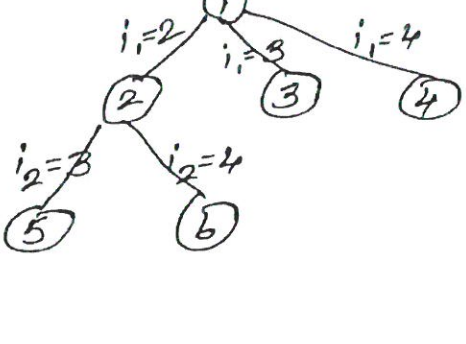
cost of node 4 is $\hat{c}(4) = \hat{c}(1) + A_1[1,4] + r$

$= 11 + 0 + 5 = 16$

The portion of the state space tree is



Among the nodes ②, ③, ④ node ② is having least cost. So expand ②



11.b) Explain the classes of NP hard and NP complete (5M)

NP hard -2M
NP complete-2M
Relation between NP hard and NP complete-1M

<u>class P:</u> class P is the set of all decision problems solvable by deterministic algorithms in polynomial time.

<u>Class NP:</u> class NP is the set of all decision problems solvable by non-deterministic algorithms in polynomial time.

<u>Class NP-complete:</u>

A decision problem is in NP-complete if it is in NP and every problem in NP is reducible to it in polynomial time.

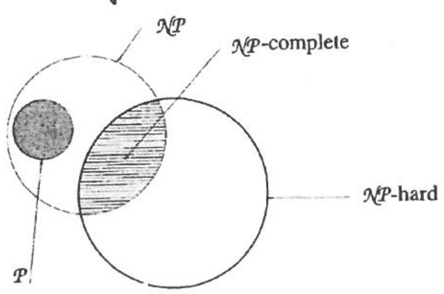
→ A NP-complete problem is solved by a polynomial time deterministic algorithm if and only if all other NP-complete problems are solved by a polynomial time deterministic

<u>Class NP-Hard:</u>

A decision problem is in NP-hard if every problem in NP is reducible to it in polynomial time.

→ If a NP-hard problem is solved by a polynomial time deterministic algorithm, then all other NP-Complete problems are solved by a polynomial time deterministic algorithm.

Relationship among P, NP, NP-complete NP-hard classes is



....

path 1-2-3:

set 2^{nd}, 3^{rd} col of A_2 to ∞

set $A_2[3,1] = \infty$

Then the resultant matrix is $A_5 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 1 & \infty & \infty & \infty \end{bmatrix}$

Reduce row 4 by 1

$$A_5 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 1 & \infty & \infty & \infty \end{bmatrix} \begin{matrix} \\ \\ \\ -1 \end{matrix} \Longrightarrow A_5 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & \infty \end{bmatrix}$$

cost of node ⑤ is $\hat{c}(5) = \hat{c}(2) + A_2[2,3] + r$

$= 15 + 0 + 1 = 16$

path 1-2-4:

set 2^{nd}, 4^{th} col of A_2 to ∞

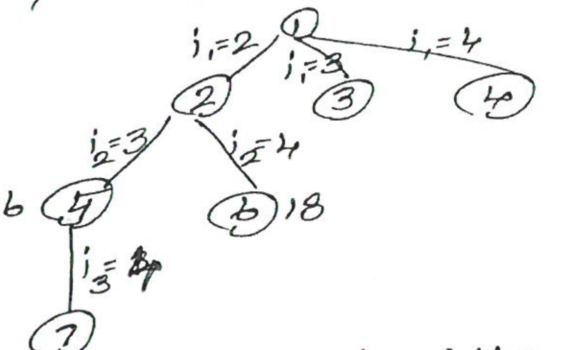
set $A_2[4,1] = \infty$

Then the resultant matrix is $A_6 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty \\ \infty & 0 & 0 & \infty \end{bmatrix}$

cost of node 6 is $\hat{c}(6) = \hat{c}(2) + A_2[2,4] + r$

$= 15 + 3 + 0 = 18$

Among nodes ⑤, ⑥, node ⑤ has least cost so expand it



For node ⑤, the only child node is node ⑦

we set the path as 1 — 2 — 3 — 4 — 1

The cost of this tour is $\hat{c}(7) = 16$

....