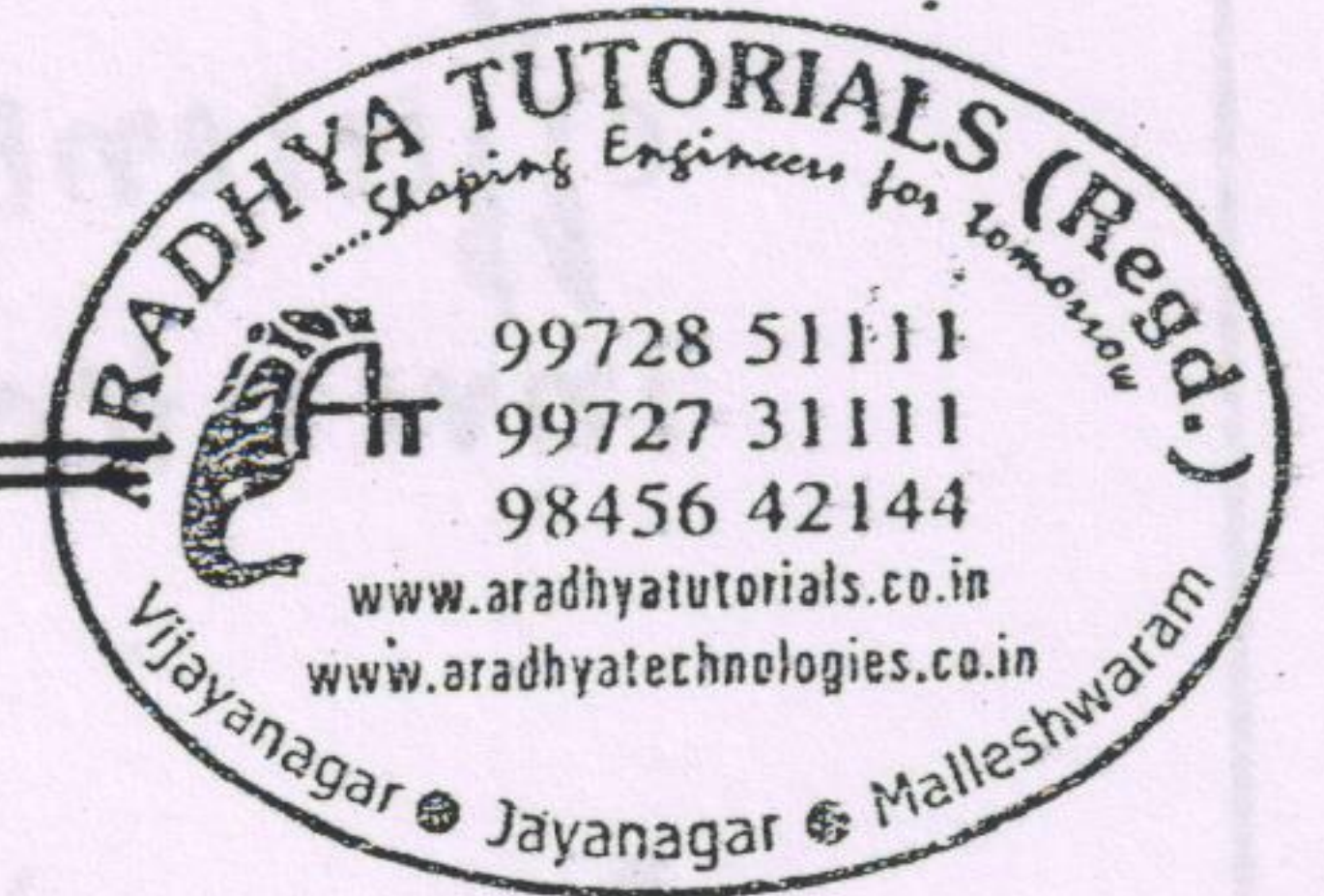


UNIT - IV

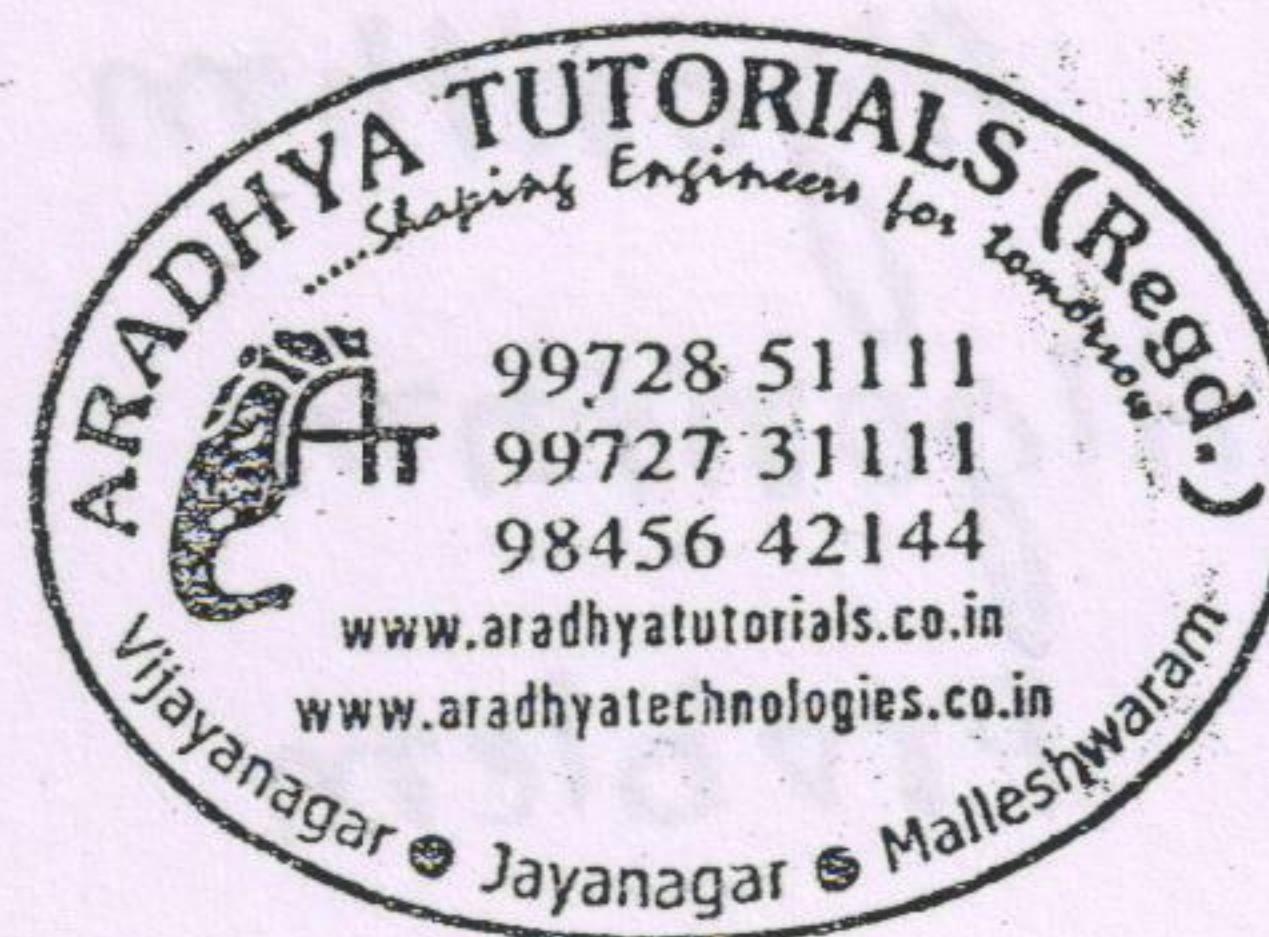


DYNAMIC PROGRAMMING

- The general method
- Warshall's Algorithm
- Floyd's Algorithm for all pair shortest path
- Single source shortest paths : general weights
- 0/1 Knapsack
- Travelling Salesperson problem

This Notes is valid
 ★ only for the duration ★
 Jan-2014 to Dec-2014

CLASSES ARE ALSO CONDUCTED
 FOR EC / EEE / IT / TC BRANCH
 SUBJECTS • ARADHYA TUTORIALS



ABC for Java and Testing
 Aradhya's Brilliance Centre for java and testing is a sister concern of Aradhya tutorials. We provide quality industry oriented training for final year students and freshers. We also provide immense job opportunities. We have already placed numerous students into the software industry. To know more, like our Facebook page "ABC For Jaya and testing". Also visit our website "www.abcforjava.org".

What is Dynamic Programming ? Explain

Dynamic Programming is one of the most efficient programming techniques. It was invented by Richard Bellman.

Dynamic Programming is used to mainly solve such problems which have overlapping sub-problems. It makes use of a recurrence relation.

Dynamic Programming suggests solving each of the smaller subproblems only once and records the results in a table from which we can then obtain a solution to the original problem.

This Notes is valid
★ only for the duration ★
Jan-2014 to Dec-2014

One variation of the dynamic programming approach seeks to avoid solving un-necessary subproblems by using the concept of memory functions.

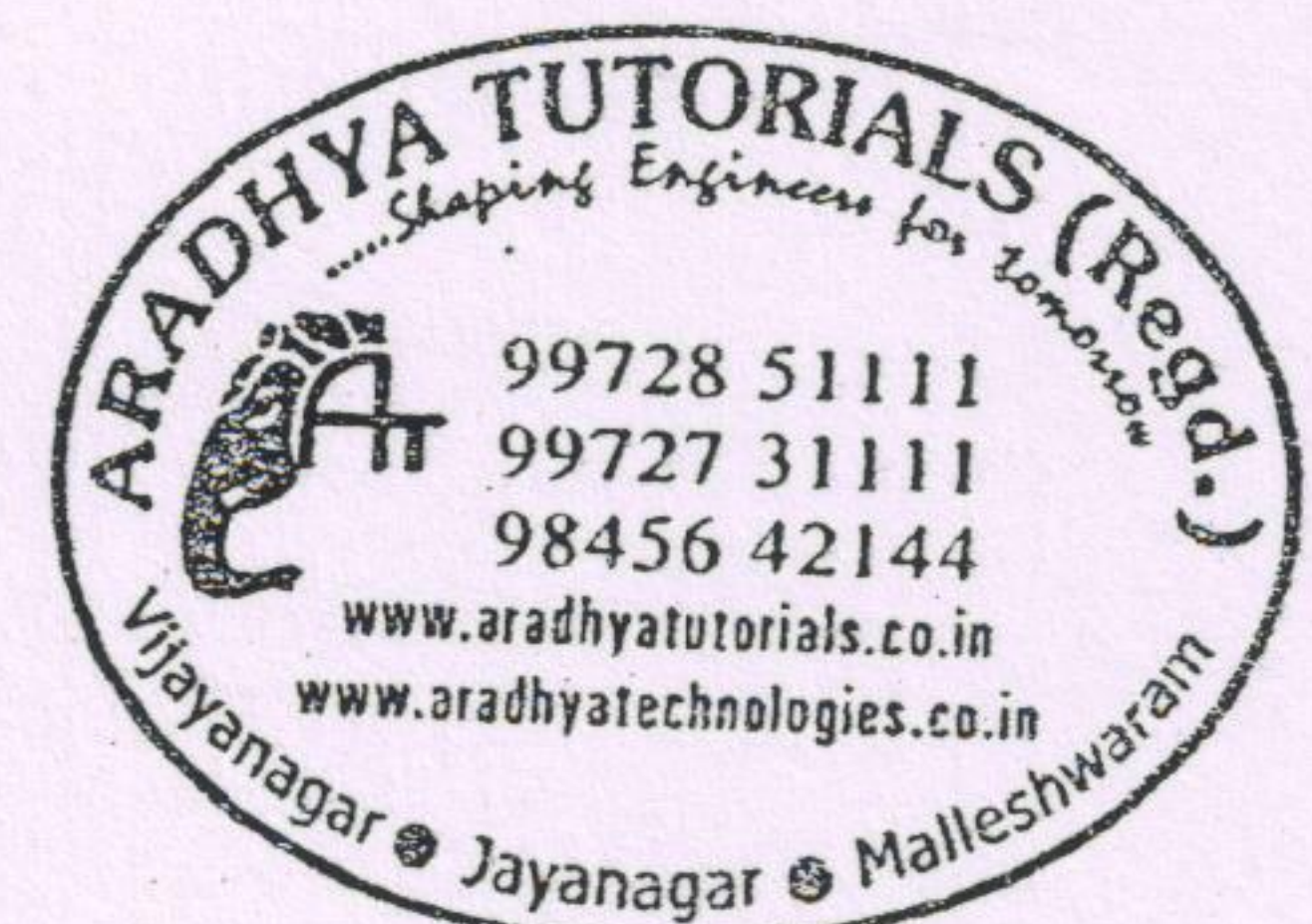
Some of the problems that can be solved using Dynamic Programming technique is -

• Warshall's Algorithm

• Floyd's Algorithm

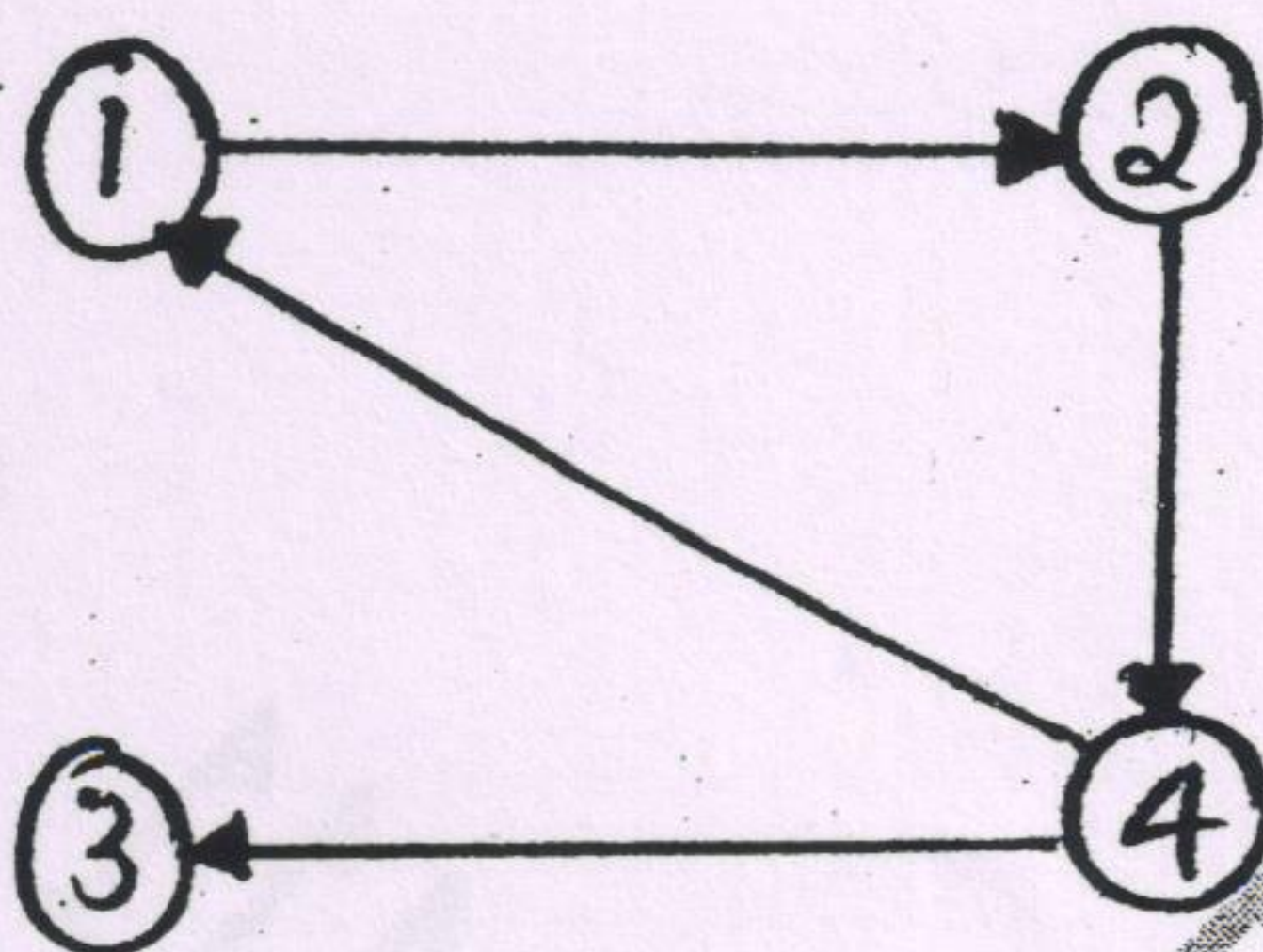
• Knapsack Problem

etc.



Write a note on Warshall's Algorithm.

Warshall's Algorithm makes use of the concept of Dynamic Programming. It is used to compute the transitive closure of the given graph. It accepts adjacency matrix as its input and generates path matrix as its output. Consider the graph



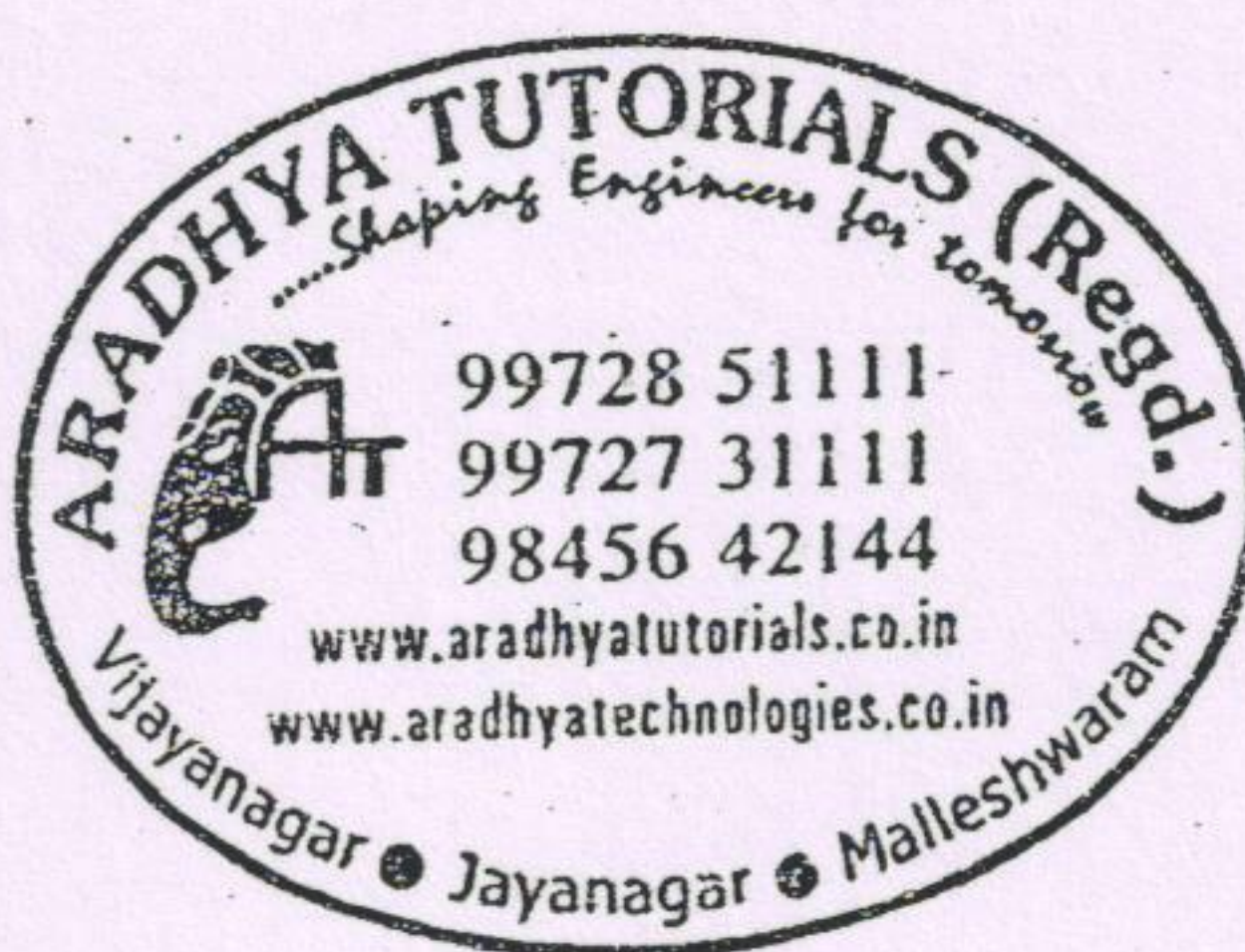
	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	0	0	0	0
4	1	0	1	0

Warshall's Algorithm

	1	2	3	4
1	1	1	1	1
2	1	1	1	1
3	0	0	0	0
4	1	1	1	1

adjacency matrix

Path matrix



The formula for computing transitive closure is as given below

$$r_{ij}^k = r_{ij}^{k-1} \text{ OR } (r_{ik}^{k-1} \text{ and } r_{kj}^{k-1})$$

This Notes is valid
★ only for the duration ★
Jan-2014 to Dec-2014

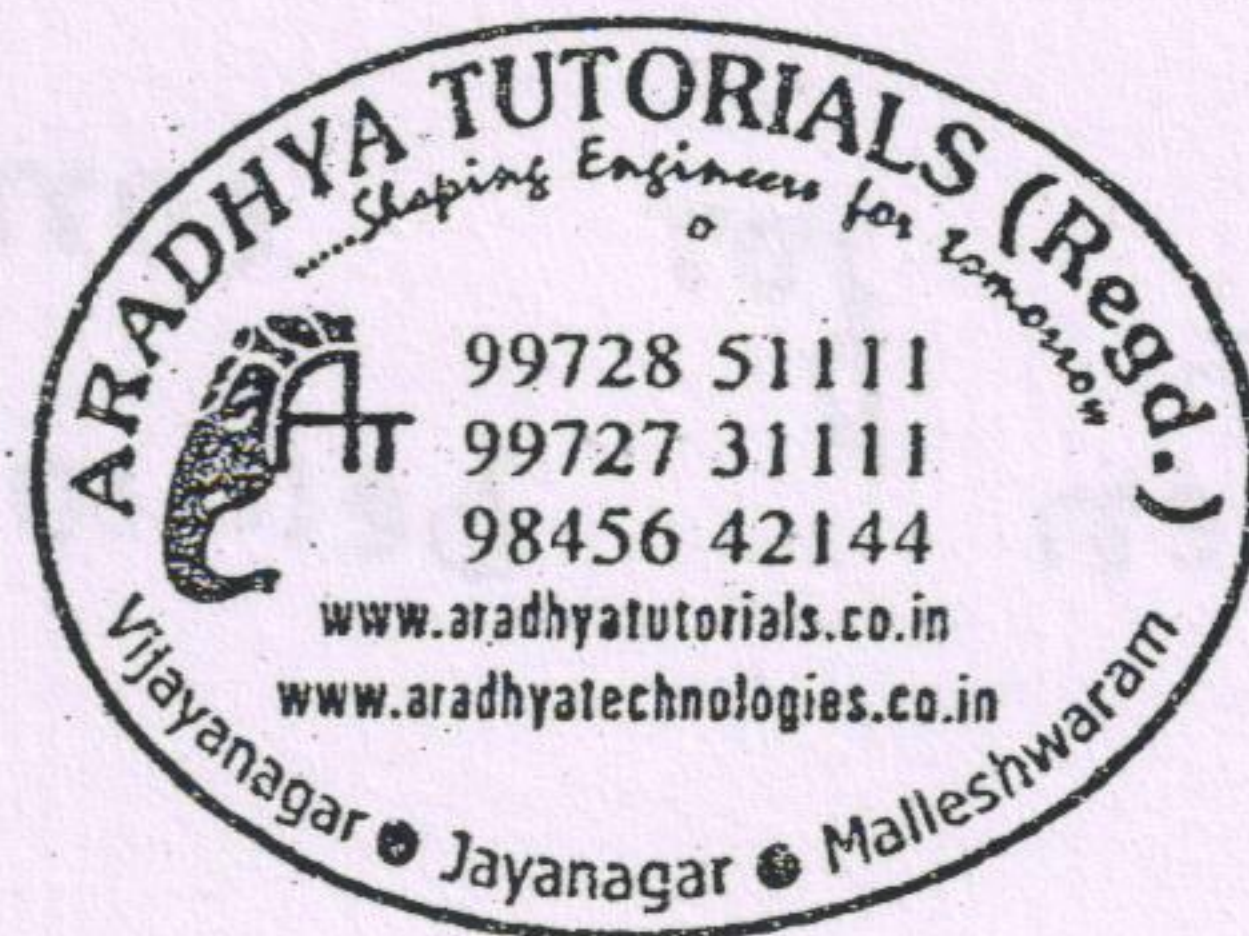
The algorithm is as given below-

This Notes is valid
★ only for the duration ★
Jan-2014 to Dec-2014

TRACE - Refer class notes

ANALYSIS - Refer class notes

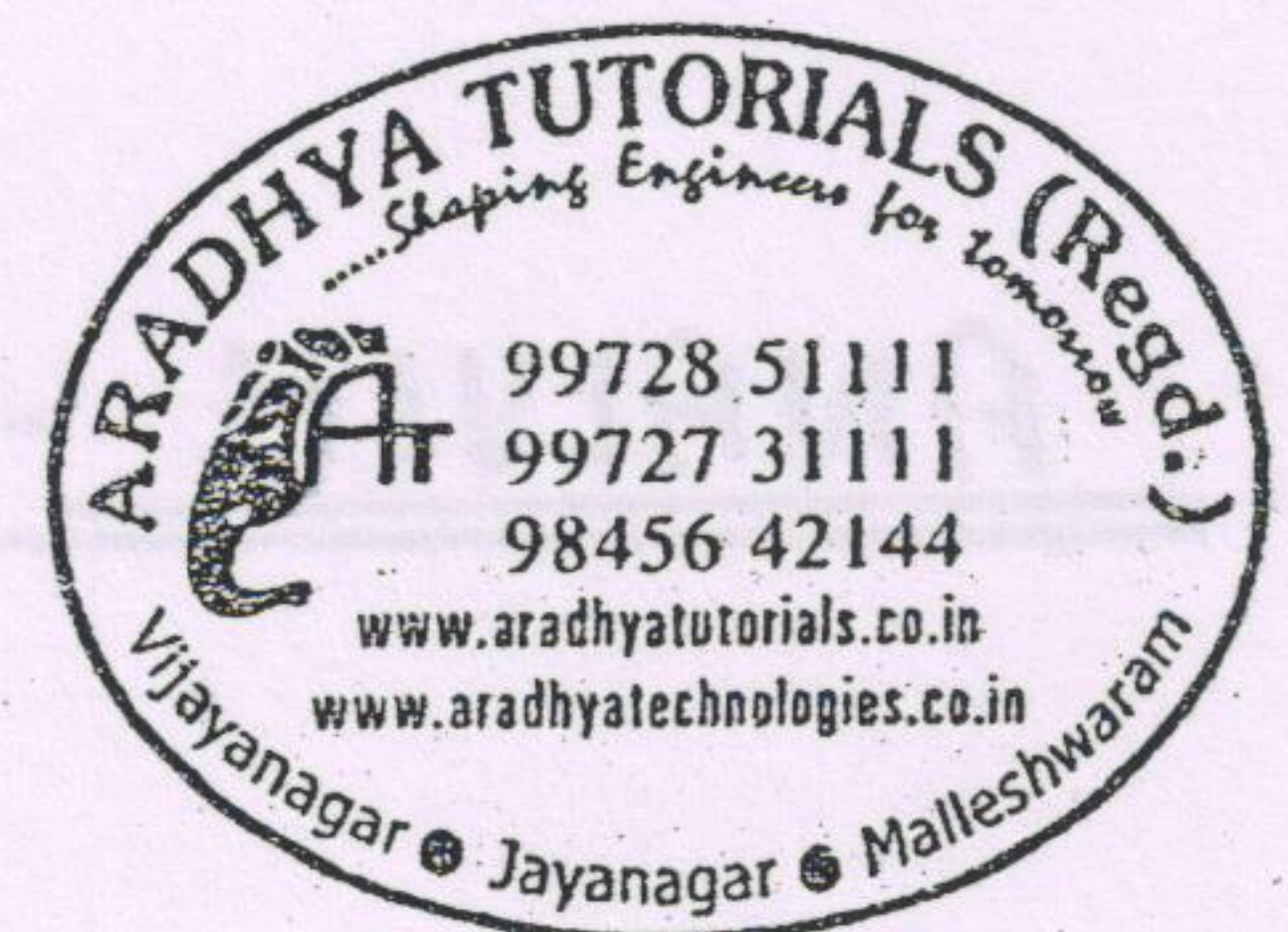
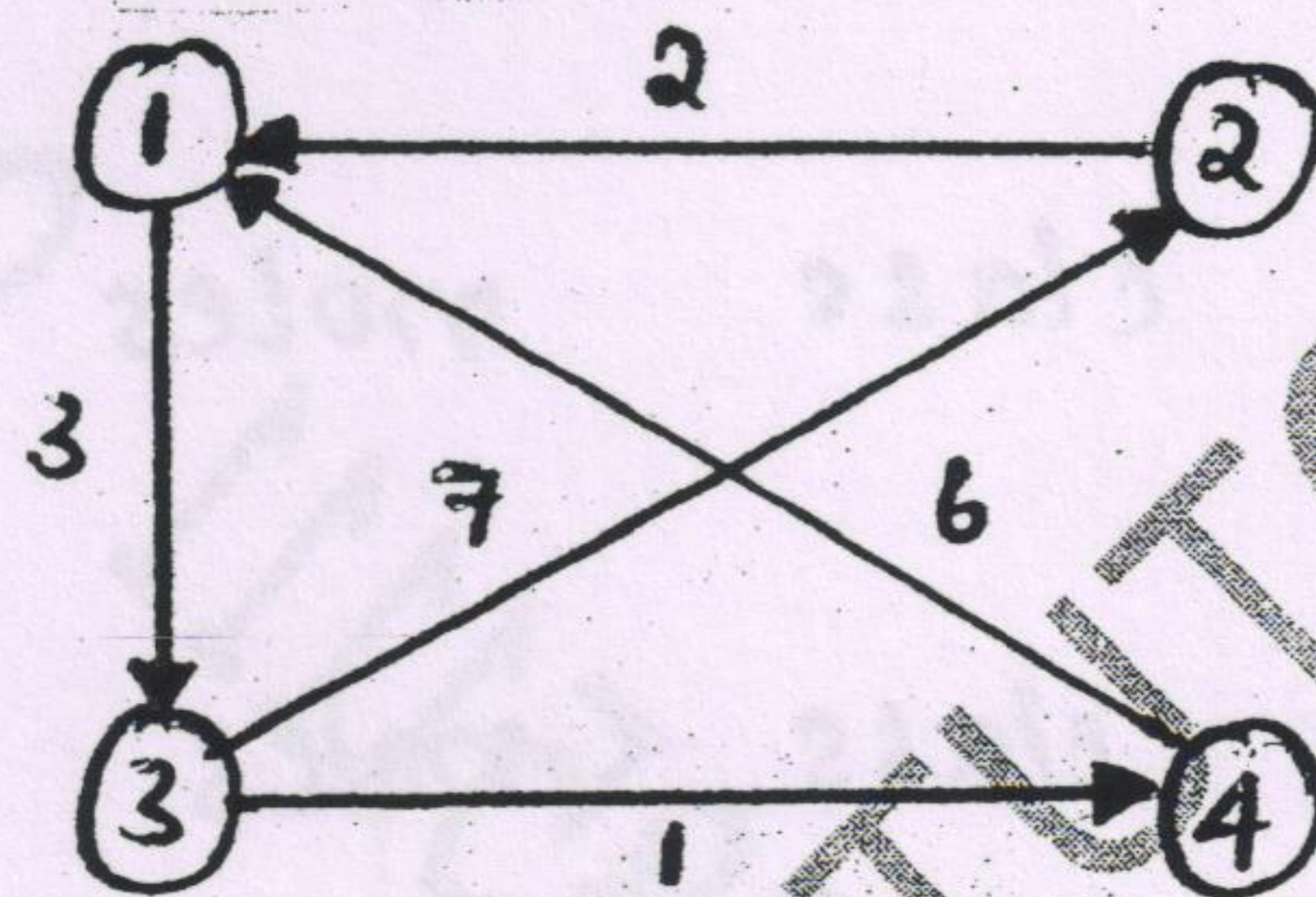
CLASSES ARE ALSO CONDUCTED
FOR EC / EEE / IT / TC BRANCH
SUBJECTS @ ARADHYA TUTORIALS



Write a note on Floyd's Algorithm.

Floyd's Algorithm makes use of the concept of Dynamic Programming. It is used to compute the shortest distance between every a pair of nodes in a given graph and hence it is also called as the "all-pair-shortest-path" algorithm. Consider the graph

This Notes is valid
 ★ only for the duration ★
 Jan-2014 to Dec-2014



	1	2	3	4
1	0	∞	3	∞
2	∞	0	∞	∞
3	∞	7	0	1
4	6	∞	∞	0

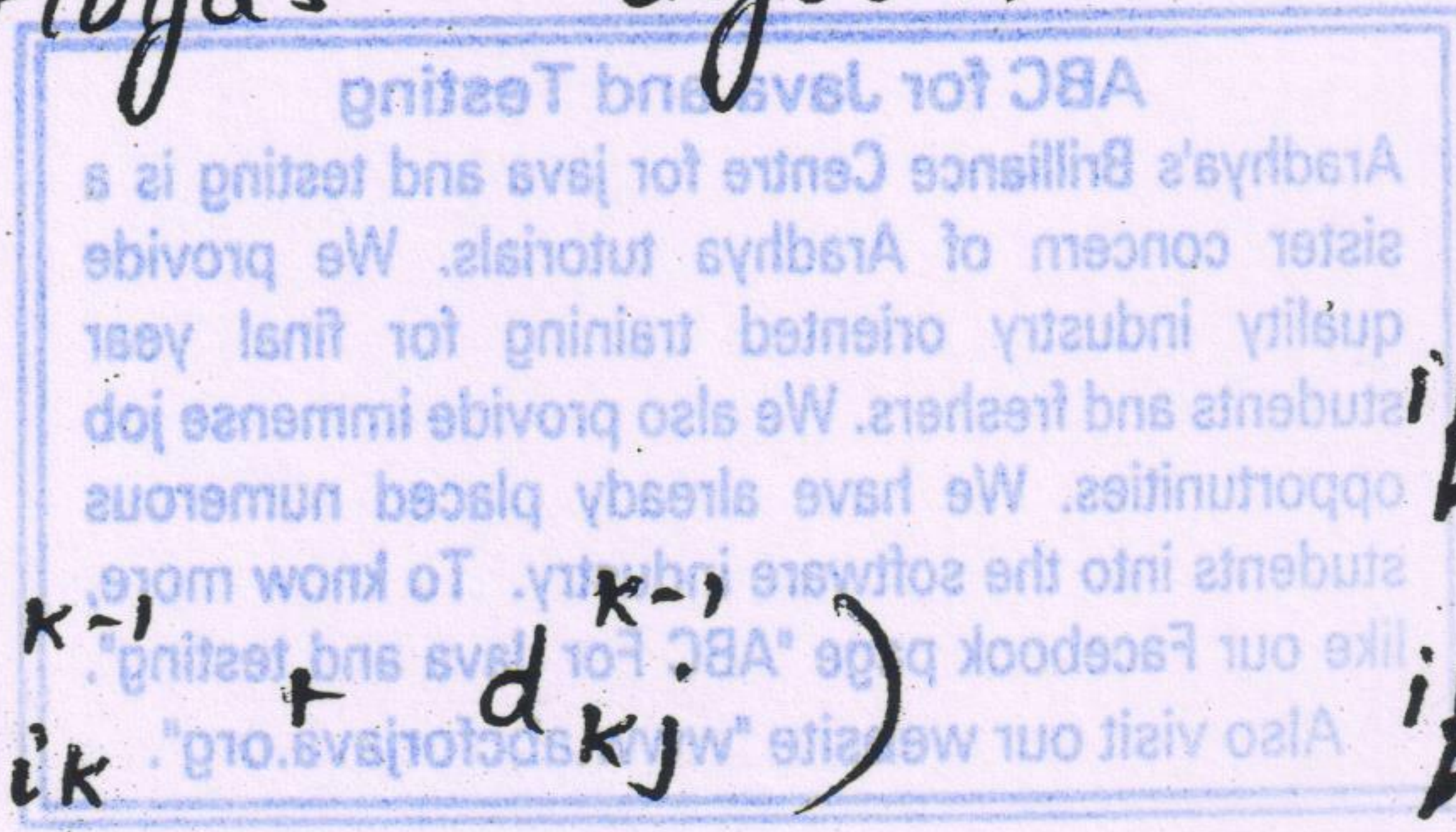
FLOYD'S ALGORITHM

	1	2	3	4
1	0	10	3	4
2	2	0	5	6
3	7	7	0	1
4	6	16	9	0

The formula for Floyd's algorithm is -

$$d_{ij} = w_{ij}$$

$$d_{ij}^k = \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}) \quad \text{if } k \geq 1$$



Floyd's Algorithm is as shown below -

This Notes is valid
★ only for the duration ★
Jan-2014 to Dec-2014

TRACE - Refer class notes

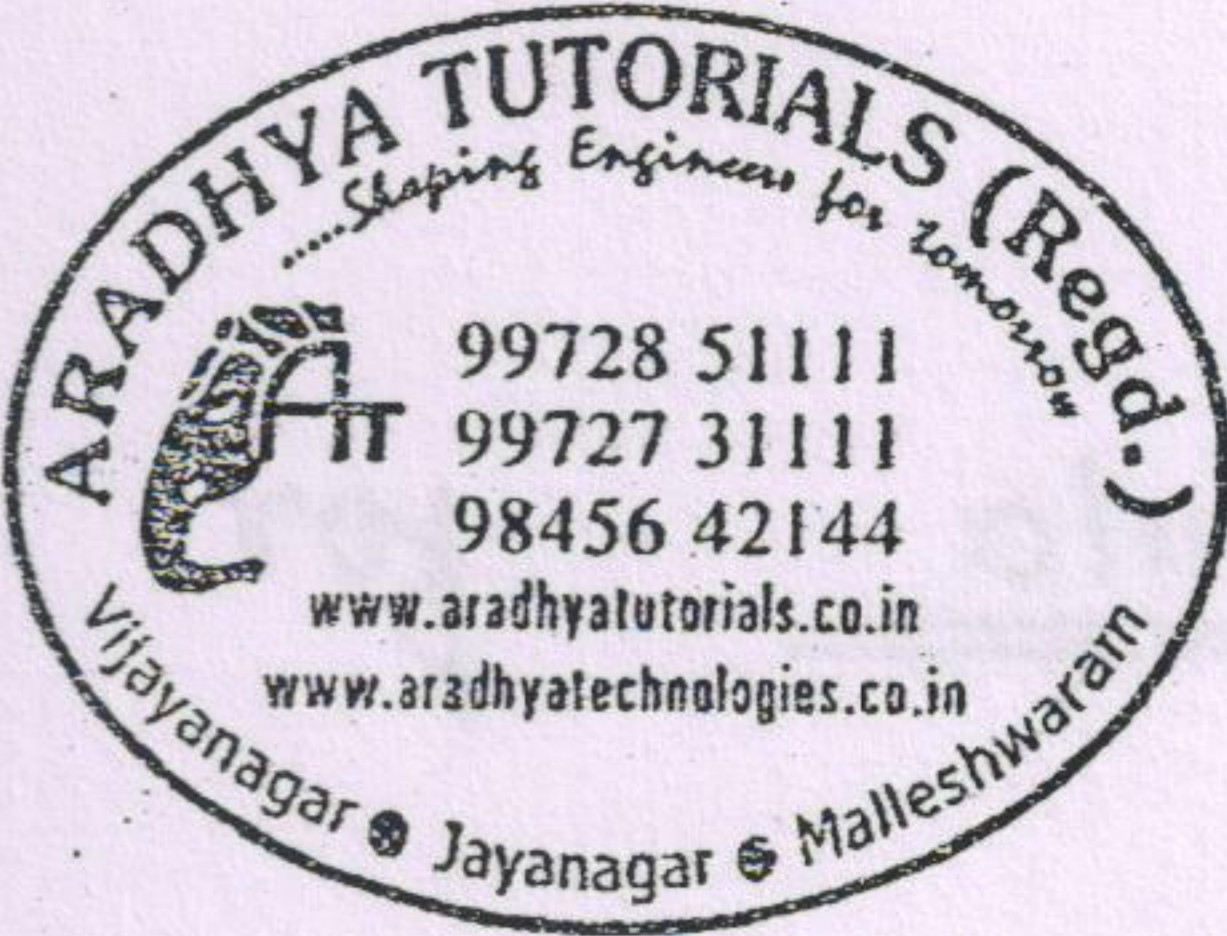
ANALYSIS - Refer class notes

9845642144

9972851111 9972731111

CLASSES ARE ALSO CONDUCTED
FOR EC / EEE / IT / TC BRANCH
SUBJECTS @ ARADHYA TUTORIALS

ABC for Java and Testing
Aradhya's Brilliance Centre for java and testing is a sister concern of Aradhya tutorials. We provide quality industry oriented training for final year students and freshers. We also provide immense job opportunities. We have already placed numerous students into the software industry. To know more, like our Facebook page "ABC For Java and testing". Also visit our website "www.abcforjava.org".



Write a note on the Knapsack Problem.

Knapsack Problem can be solved by making use of the Dynamic programming approach.

Knapsack problem is such a problem in which we have "n" items and the weights of each item is " w_1, w_2, \dots, w_n " and the values associated with each item is " v_1, v_2, \dots, v_n ". We must find the most valuable set of items that fit into the knapsack.

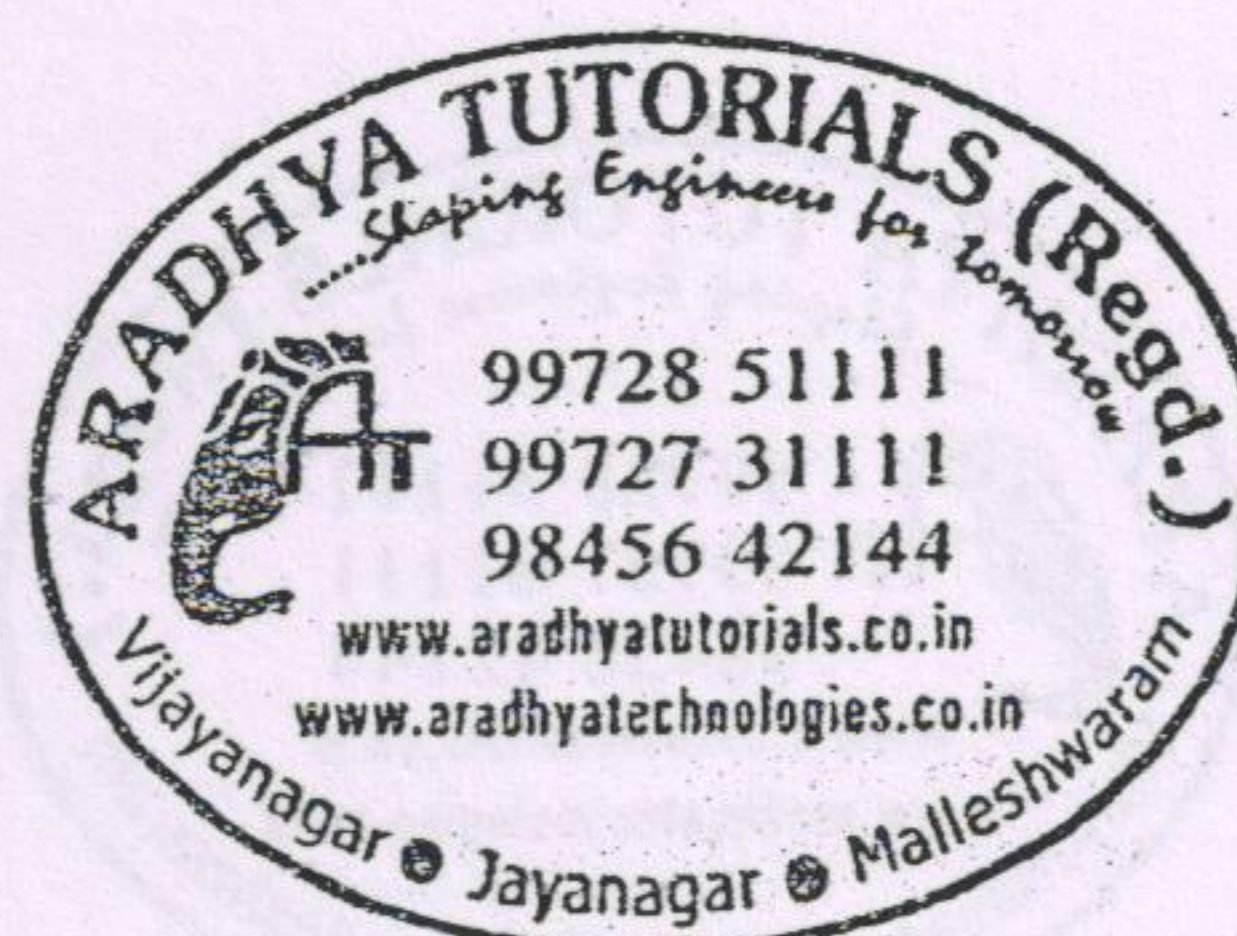
The recurrence relation related to the Knapsack problem is -

CLASSES ARE ALSO CONDUCTED FOR EC / EEE / IT / TC BRANCH SUBJECTS @ ARADHYA TUTORIALS

$$v_{ij} = \begin{cases} v_{i-1, j} & \text{if } j - w_i < 0 \\ \max[v_{i-1, j}, (p_i + v_{i-1, j-w_i})] & \text{if } j - w_i \geq 0 \end{cases}$$

This Notes is valid
★ only for the duration ★
Jan-2014 to Dec-2014

Also visit our website "www.aradhyatutorials.org"
like our Facebook page "ABC For Java and Testing"
students into the software industry. To know more
opportunities. We have already placed numerous
students and freshers. We also provide immense job
quality industry oriented training for final year
sister concern of Aradhyas tutorials. We provide
Aradhyas' Brilliance Centre for Java and Testing is a
ABC for Java and Testing



P.T.O

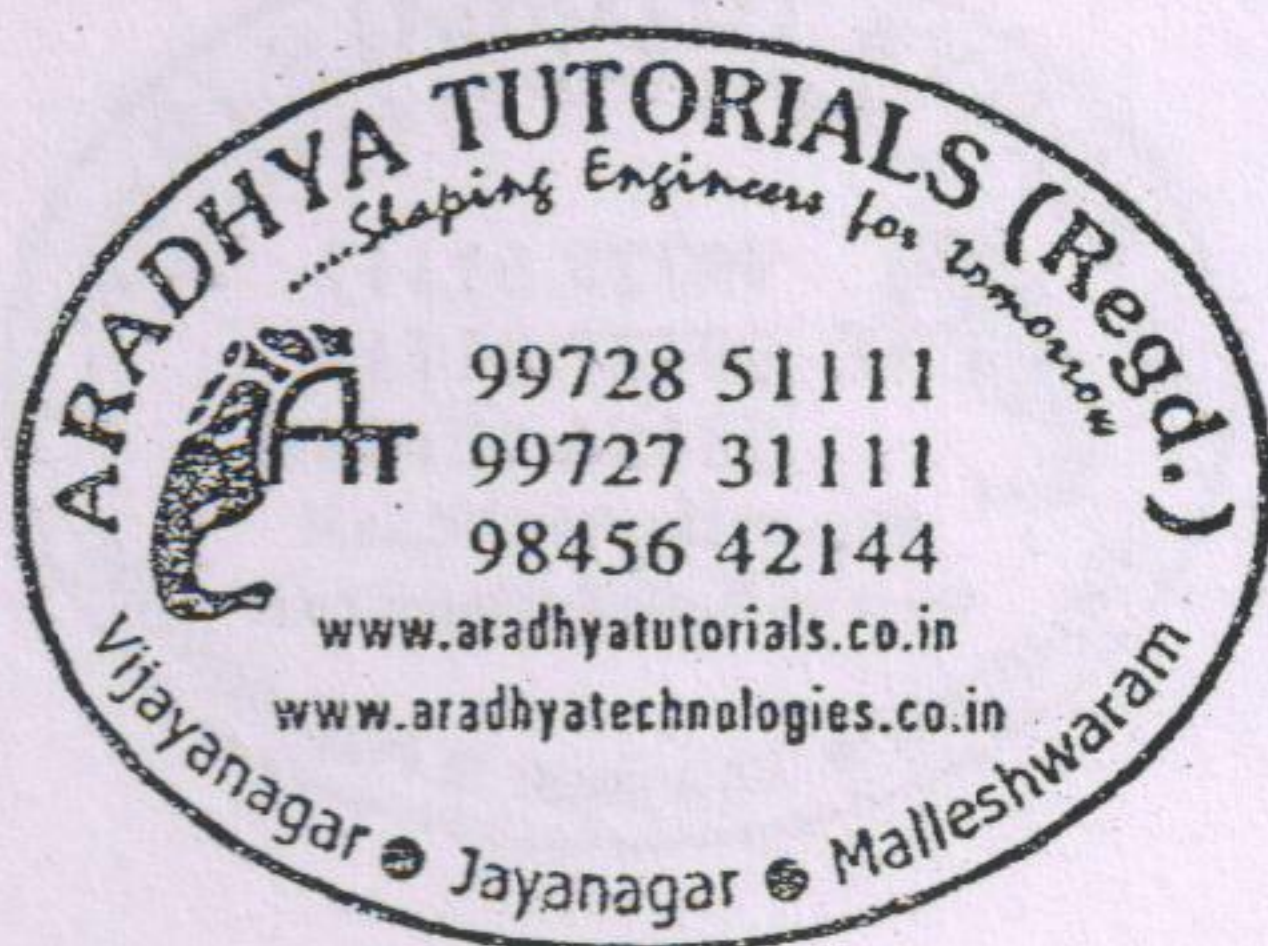
The Algorithm for Knapsack is as shown below -

This Notes is valid
★ only for the duration ★
Jan-2014 to Dec-2014

TRACE - Refer class notes

ANALYSIS - Refer class notes

CLASSES ARE ALSO CONDUCTED
FOR EC / EEE / IT / TC BRANCH
SUBJECTS @ ARADHYA TUTORIALS



ABC for Java and Testing
Aradhya's Brilliance Centre for java and testing is a sister concern of Aradhya tutorials. We provide quality industry oriented training for final year students and freshers. We also provide immense job opportunities. We have already placed numerous students into the software industry. To know more, like our Facebook page "ABC For Java and testing". Also visit our website "www.abcforjava.org".