	PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043
	Department of Information Technology **S.No.-27, Pune Satara Road, Dhankawadi, Pune-411043**

QB for Design and Analysis of Algorithm

Unit-I

1.	Reorder the following complexities from the smallest to the largest [5] i $n\log_2 n$, $n + n^2 + n^3$, 24, sqrt(n). ii $n!$, 2^n, $(n + 1)!$, 2^{2n}, n^n, $n^{\log n}$
2.	Prove by contradiction that the square root of 2 is irrational.
3.	Explain the potential method of amortized analysis with an example.
4.	Solve the following recurrence relation using substitution method $T(n) = T(n - 1) + 1$, $T(0) = 0$.
5.	Solve the following Recurrence relation using substitution method and write the time complexity. $T(n) = 2\,T(n/2) + n \quad n>1$ $T(n) = 1$ if $n=1$
6.	Find Brute force solution to 8 queen's problem.
7.	Prove by Mathematical Induction that for each positive number n $1+2+3+ \ldots\ldots\ldots\ldots +n=n(n+1)/2$.
8.	State Advantages and Disadvantages of Brute Force Method
9.	Write an Algorithm and recurrence relation with the initial condition of "Tower of Hanoi Problem". Find its time efficiency using backward substitution method.
10.	Write an algorithm to solve 8 queen's problem using Brute force method

QB for Design and Analysis of Algorithm

11.	List the properties of various asymptotic notations.
12.	Describe the strategy to analyze the recursive algorithm with suitable examples
13.	Describe the strategy to analyze the non-recursive algorithm with suitable examples.
14.	Compare apriori & posteriori analysis of algorithms.
15.	Prove by mathematical induction that "the sum of cubes of n integers is equal to the square of sum of these integers"
16.	Solve the following recurrence relation using the substitution method. $T(n) = 2T(\sqrt{n}) + C \quad n>2$ $= 1 \quad n<=2$
17.	Explain the difference between a direct proof and a proof by contradiction. Provide an example of each in the context of algorithm analysis
18.	What is mathematical induction, and how is it used to prove the correctness of algorithms?
19.	How is a proof by contraposition different from a direct proof? Provide an example related to algorithm design.
20.	How would you use loop invariants to prove the correctness of an iterative algorithm? Provide an example with the Insertion Sort algorithm
21.	How does the brute force method work in pattern matching, and what are its time complexity and practical limitations compared to other pattern matching algorithms?
22.	What is the brute force method for pattern matching, and how does it work in searching for a specific pattern within a larger text?

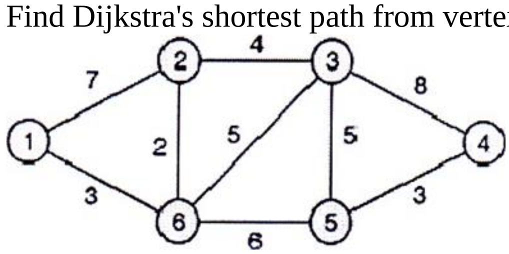
PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043

Department of Information Technology

S.No.-27, Pune Satara Road, Dhankawadi, Pune-411043

QB for Design and Analysis of Algorithm

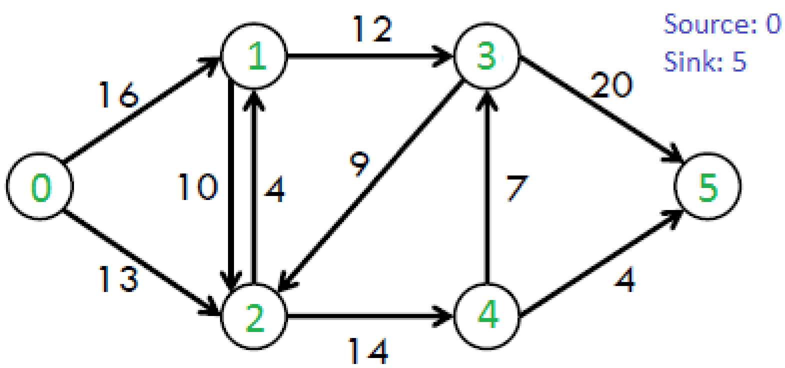
Unit-II

1.	What is the divide and conquer method? Explain control abstraction algorithm of divide and conquer method.
2.	Write down the algorithm for binary search and solve the recurrence relation for it using the substitution method.
3.	Write a recurrence relation for Merge sort and Find a time complexity using Master's theorem.
4.	Write an algorithm to find the Minimum Spanning Tree using Kruskal algorithm and analyze it.
5.	Write a recurrence relation for Quicksort in the worst and best case and also solve that by using Master's theorem.
6.	Write a recursive algorithm for finding maximum and minimum using divide and conquer and verify its time complexity
7.	How do Prim's and Kruskal's algorithms differ in constructing a minimum spanning tree, and what are their respective time complexities?
8.	Find Dijkstra's shortest path from vertex 1 for the following graph 
9.	Solve the following job sequencing with deadlines problem. n = 7, Profits (p1,p2,........,p7)={3,5,20,18,1,6,30} Deadlines(dl,d2,.....,d7)={1,3,4,3,2,1,2}.
10	Write a recurrence relation of the best-case of quick sort and determine its time complexity.

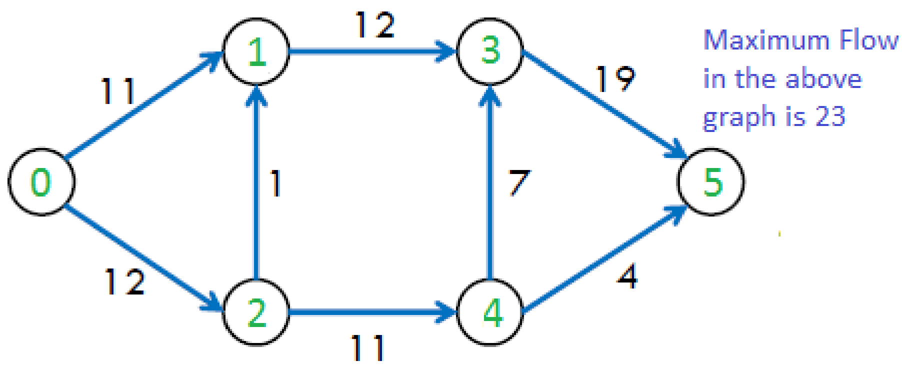
QB for Design and Analysis of Algorithm

11 Given items I= (I1,I2,I3,I4,I5), Weight w=(5,10,20,30,40) and Profit p=(30,20,100,90,160). Let us consider that the capacity of the knapsack *W* **= 60. Find the maximum profit using fractional knapsack**

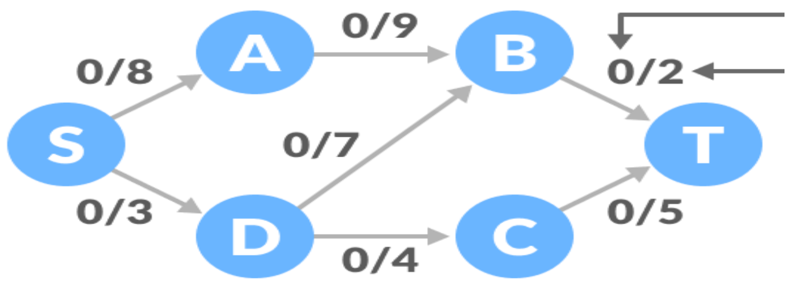
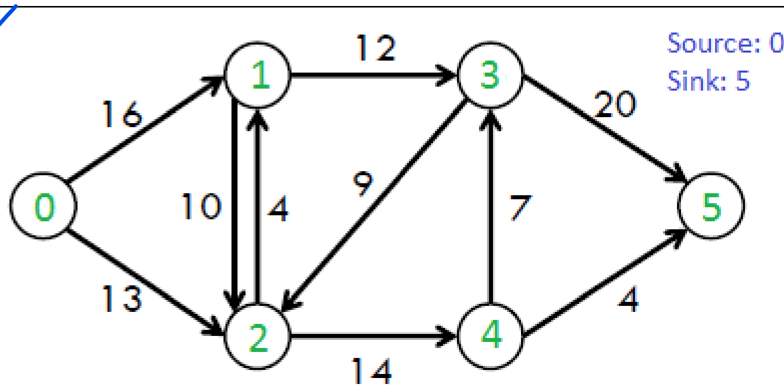
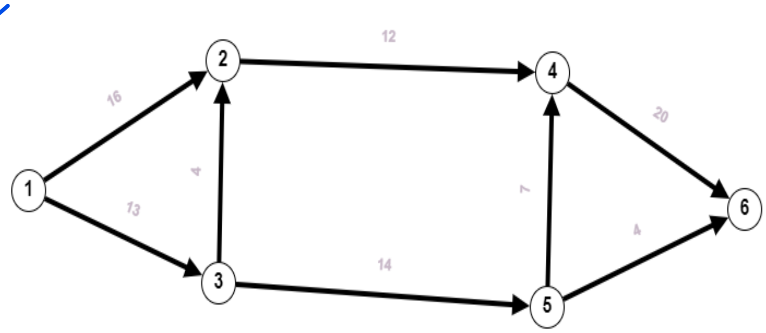
12



The maximum possible flow in the above graph is 23.

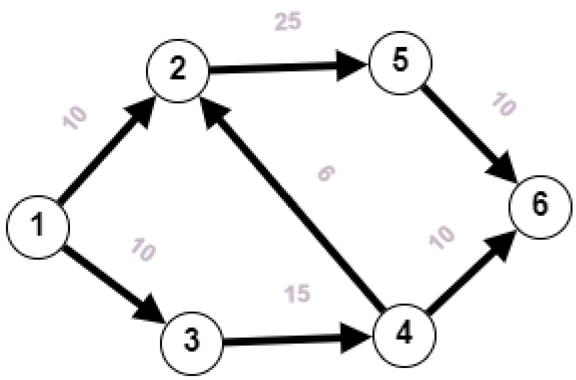


QB for Design and Analysis of Algorithm

13	
14	
15	

QB for Design and Analysis of Algorithm

16



17 Given the jobs, their deadlines and associated profits as shown, what is the maximum earned profit? J=6, D= (5,3,3,2,4,2) P= (200,180,190,300,120,100)

18 consider the following tasks with their deadlines and profits. Schedule the tasks in such a way that they produce maximum profit after being executed –

S. No.	1	2	3	4	5
Jobs	J1	J2	J3	J4	J5
Deadlines	2	2	1	3	4
Profits	20	60	40	100	80

19 Find the optimal solution for the fractional knapsack problem making use of greedy approach. Consider- n = 5 w = 60 kg (w1, w2, w3, w4, w5) = (5, 10, 15, 22, 25) and (b1, b2, b3, b4, b5) = (30, 40, 45, 77, 90)

20 Consider the following instance of the knapsack problem: W :15, n :7
Profit (P): (5 10 15 7 8 9 4)
Weight(w): (1 3 5 4 1 3 2) Find the optimal solution for the fractional knapsack problem making use of greedy approach.

21 Consider the following instance of the knapsack problem: n = 3, m = 20, (p1, p2, p3) = (25, 24, 15) and (w1, w2, w3) = (18, 15, 10). Find the optimal solution for the fractional knapsack problem making use of greedy approach

QB for Design and Analysis of Algorithm

Unit-III

1. State and explain the principle of Optimality.

2. Solve the knapsack problem using Dynamic programming for no. of objects n = 4, given capacity M = 8

Items	l	2	3	4
Value	l5	l0	9	5
Weight	l	5	3	4

3. Write a Bellman Ford algorithm to find the shortest path and analyze it.

4. Solve the knapsack problem using Dynamic programming for no. of objects n = 3, given capacity M = 6

Items	l	2	3
Value	10	L5	40
Weight	l	2	3

5. Solve the knapsack problem using Dynamic programming for no. of objects n = 4, given capacity M = 8

Items	1	2	3	4

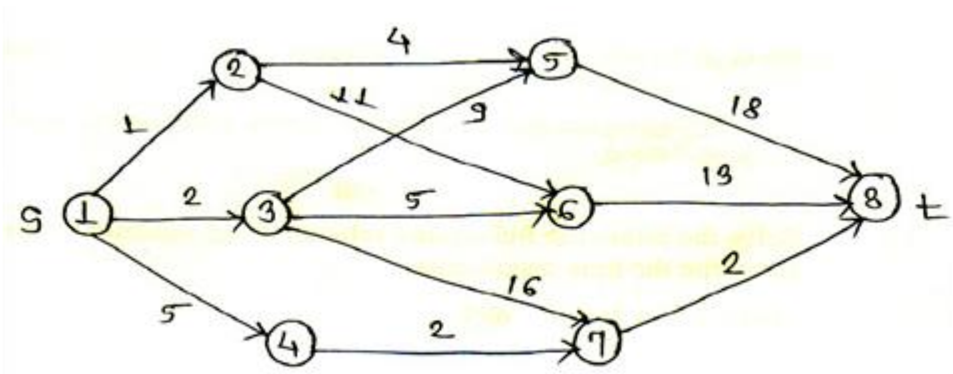
QB for Design and Analysis of Algorithm

Weights	3	4	6	5
Value	2	3	1	4

6. Solve the traveling salesman problem with associated cost adjacency matrix using dynamic programming.

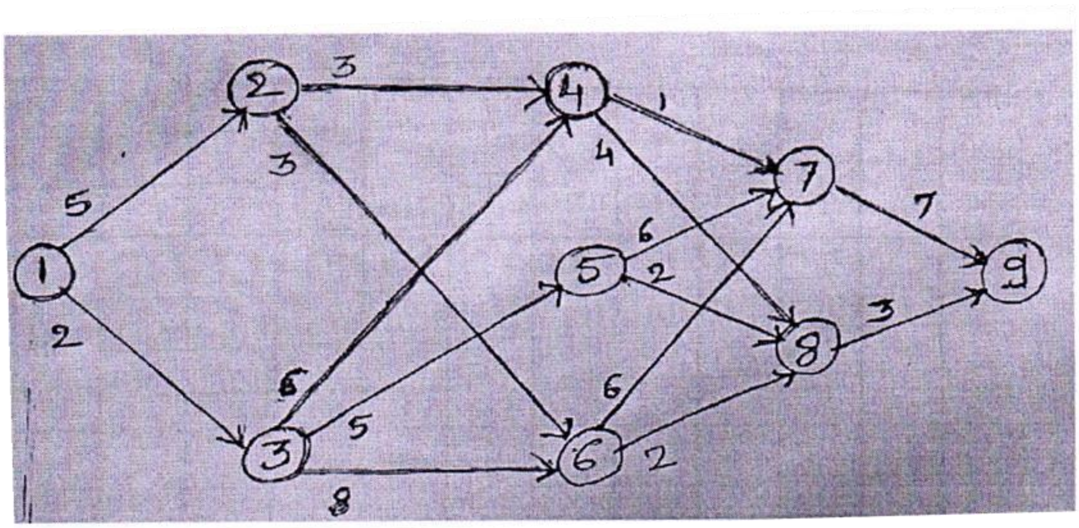
	A	B	C	D
A	0	4	2	1
B	4	0	13	9
C	2	13	0	8
D	1	9	8	0

7. Find minimum cost path from source (s) to sink (t) of the following multistage graph.[8]

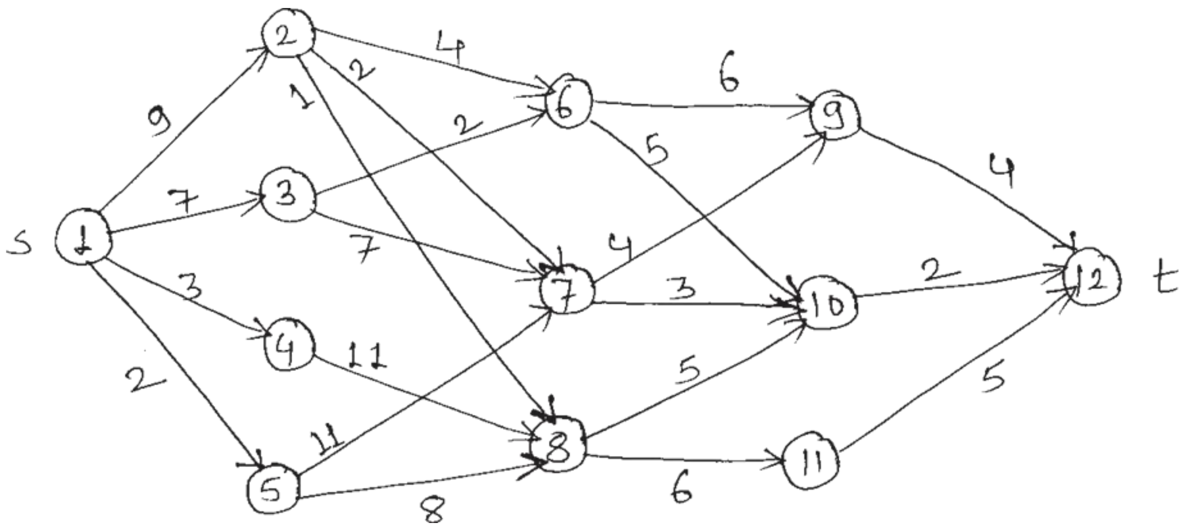


8. Find minimum cost path from source (1) to sink (t) of the above multistage graph.

QB for Design and Analysis of Algorithm

	
9.	Compare dynamic programming with divide & Conquer.
10.	Compare dynamic programming with greedy approach
11.	Find the solution of following a travelling salesman problem using dynamic programming. $\text{cost matrix} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix} \end{array}$
12.	Find the minimum cost path from source (s) to sink (t) of the following multistage graph

QB for Design and Analysis of Algorithm



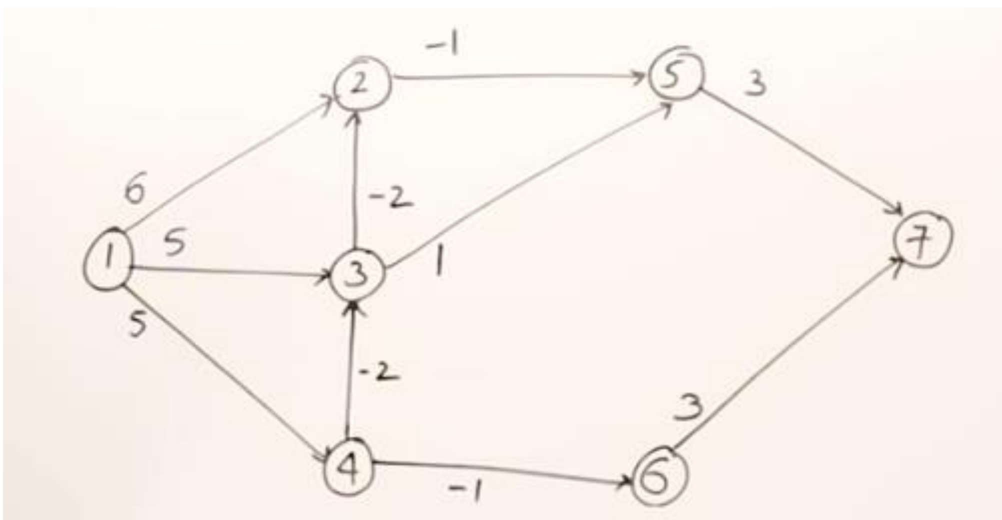
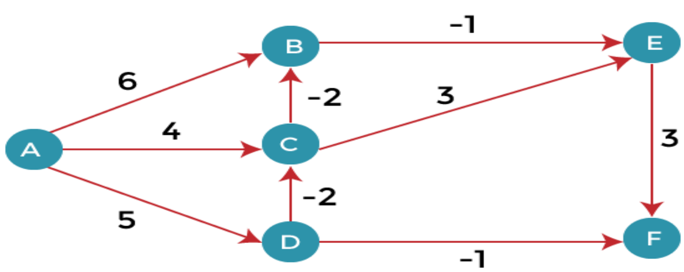
13. Find out min no. coins to make change of given amount using given coins:
Coins {1, 5, 6, 9}
W=10.

	0	1	2	3	4	5	6	7	8	9	10
1	0	1	2	3	4	5	6	7	8	9	10
5	0	1	2	3	4	1	2	3	5	5	2
6	0	1	2	3	4	1	1	2	4	4	2
9	0	1	2	3	4	1	1	2	1	1	2

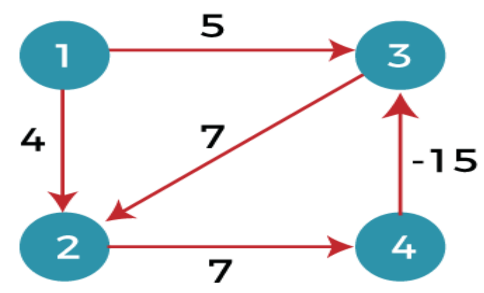
If(a[i]>j)then {a[i][j]=a[i-1][j]
Else a[i][j]=min(a[i-1][j],1+a[i][j-coins[i])

14. Find the minimum cost shortest path from source (1) to sink (7) of the following graph using bellman ford single source shortest path algorithm

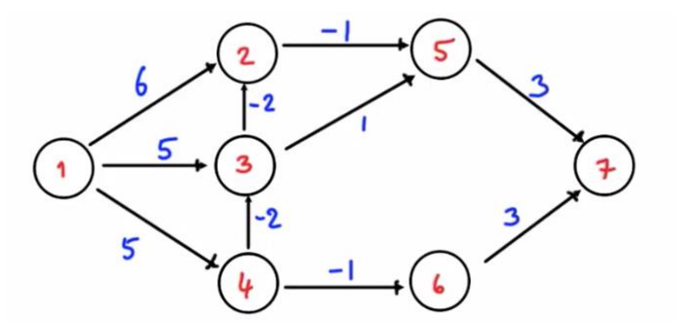
QB for Design and Analysis of Algorithm

	
15.	Find the minimum cost shortest path from source (A) to sink (F) of the following graph using bellman ford single source shortest path algorithm 
16.	Find the minimum cost shortest path from source (1) to sink (4) of the following graph using bellman ford single source shortest path algorithm

QB for Design and Analysis of Algorithm



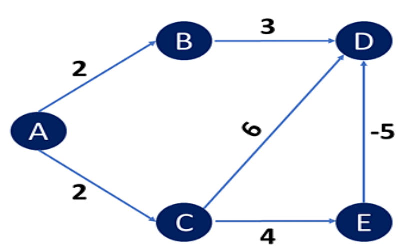
17.



$Dist^{k}$ [u]= min ($Dist^{k-1}$ [u], $Dist^{k-1}$ [w] + cost[w, u] for each neighbor w of u)

18. Write Pseudo code of Bellmon-Ford Algorithm .Also write its time complexity.

Initialize(G, s);
for i := 1 to |V[G]| –1 **do**
    for each (u, v) in E[G] **do**
        Relax(u, v, w)
    od
od;
for each (u, v) in E[G] **do**
    if d[v] > d[u] + w(u, v) **then**
        return false
    fi

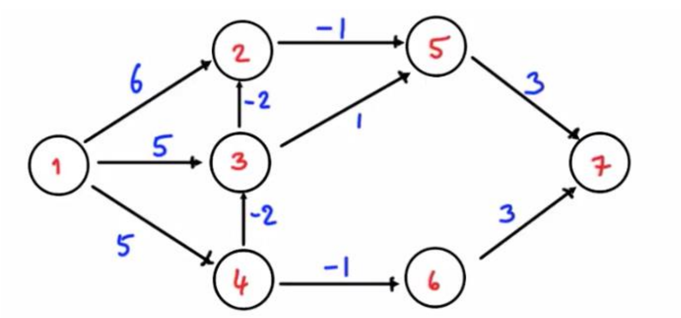
	od; **return** true
19.	Find the minimum cost shortest path from source (A) to sink (F) of the following graph using bellman ford single source shortest path algorithm 
20.	Compare Dijkstra's Algorithm and Bellman-Ford Algorithm to find Single Source Shortest Problem.
21.	coins[] = {2, 5, 3, 6} sum = 10 find total no. of solution by using dynamic programming approach. **Explanation:** There are five solutions: {2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5} and {5,5}.
22.	Find the minimum cost shortest path from source (1) to sink (8) of the following multistage graph given

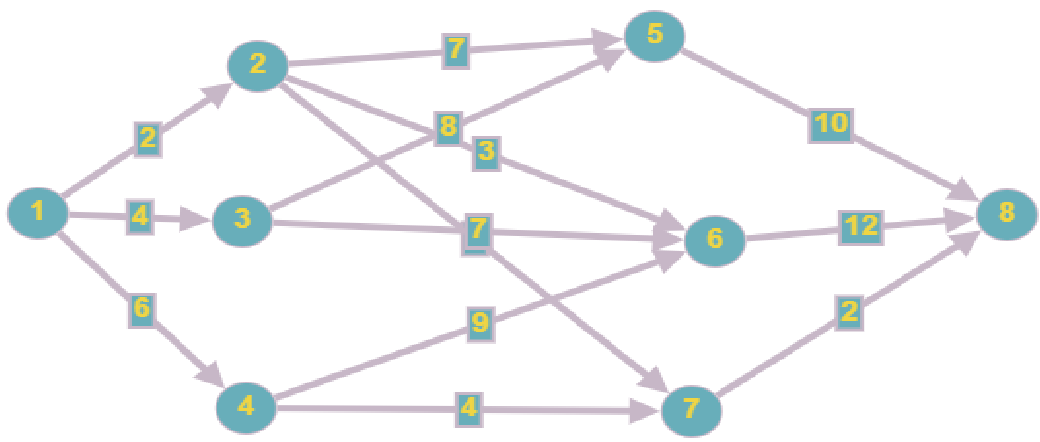
PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043

Department of Information Technology

S.No.-27, Pune Satara Road, Dhankawadi, Pune-411043

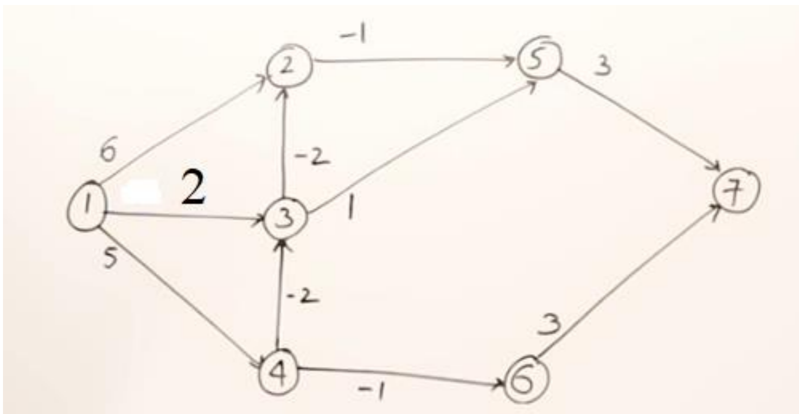
QB for Design and Analysis of Algorithm

<table>
<tr><td></td><td>




</td></tr>
<tr><td>23.</td><td>Find the minimum cost shortest path from source (1) to sink (7) of the following graph using bellman ford single source shortest path algorithm</td></tr>
</table>

QB for Design and Analysis of Algorithm

	
24.	Write the pseudocode algorithm for multistage graph problems using the Forward substitution method

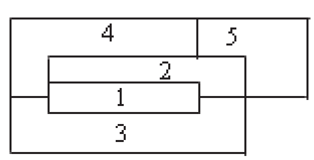
QB for Design and Analysis of Algorithm

Unit-IV

1.	Write a recursive and Iterative algorithm of backtracking method
2.	Explain the 8-Queens problem & explain the following with respect to the 8-Queens problem. i) State space tree ii) Solution State iii) State space iv) Answer state v) Static tree vi) Dynamic tree vii) Live node viii) Bounding function
3.	Let W = {5, l0, l2, l3, l5, l8} and M = 30. Find all possible subsets of W that sum to M. Draw the portion of the state space tree.
4.	Solve the 4-queen Problem. Draw the state space tree for the same
5.	Let W = {5, 7, l0, l2, l5, l8, 20} and M = 35. Find all possible subsets of W that sum to M. Draw the portion of the state space tree.
6.	Let W = {3.5, 6, 7} and M = 15. Find all possible subsets of W that sum to M. Draw the portion of the state space tree
7.	Let W = {11,13, 24,7} and M = 31. Find all possible subsets of W that sum to M. Draw the portion of the state space tree
8.	Write an algorithm for backtracking solution to the 0/l knapsack problem. **Or** Write an algorithm for 0/1 knapsack problem using backtracking method.

	PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043
	Department of Information Technology **S.No.-27, Pune Satara Road, Dhankawadi, Pune-411043**

QB for Design and Analysis of Algorithm

9.	Explain the following terms: i State space tree. ii Live node. iii E-node. iv Dead node.
10.	State the principal of backtracking and write backtracking algorithm for N-Queen problem
11.	Write a recursive backtracking algorithm for the sum of subsets problem.
12.	Write a recursive algorithm which shows a recursive formulation of the backtracking technique and explain it.
13.	If m = 30, Given data set w = {5, l0, l2, l3, l5, l8} find all possible subsets of w that sum to m. Draw the portion of the state space tree that is generated by the sum of the subset. Are there any differences in the computing time in a given set of elements? w = {l8, l5, l3, l2, l0, 5} And w = {l5, l3, 5, l8, l0, l2}
14.	Construct planar graph for following map. Explain how to find m - coloring of this planar graph by using m - coloring Backtracking algorithm. 
15.	Write an algorithm for graph coloring problem using backtracking method.
16.	Find all possible solutions for 5-Queen's problem using backtracking method.

QB for Design and Analysis of Algorithm

17.	What is m-colorability optimization problem? Write an algorithm for m-coloring of the graph.
18.	Explain the 0/1 knapsack problem. Provide solution to following problem instance using backtracking. N=5, M=12, P1-P5 =(10,15,6,8,4), W1-W5=(4,6,3,4,2)
19.	Compare between Backtracking and Branch and Bound method

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043

Department of Information Technology

S.No.-27, Pune Satara Road, Dhankawadi, Pune-411043

QB for Design and Analysis of Algorithm

Unit-V

1.	What is branch and bound? What are the different types of Brach and bound.
2.	Write an algorithm for FIFO branch and bound.
3.	Solve the following instance of 0/l knapsack problem by LC branch and bound approach N = 4, (pl, p2, p3, p4) = (l0, l0, l2, l8) (wl, w2, w3, w4) = (2, 4, 6, 9) and M = l5
4.	What is travelling salesman problem? Find the solution of the following travelling salesman problem using branch and bound method cost matrix = $\begin{bmatrix} \infty & 4 & 2 \\ 3 & \infty & 4 \\ 1 & 8 & \infty \end{bmatrix}$
5.	Explain the following terms: i Branch and bound. ii LC search. iii Bounding Function.
6.	Construct the solution of following Travelling Salesperson problem using Branch and Bound.

QB for Design and Analysis of Algorithm

	$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$
7.	Solve the following instance of 0/1 knapsack problem by FIFO branch and bound approach. N = 4, (p1, p2, p3, p4) = (10, 10, 12, 18) (w1, w2, w3, w4) = (2, 4, 6, 9) and M = 15.
8.	Write an algorithm for Least Cost (LC) branch and bound with suitable example.
9.	Differentiate between Backtracking & branch and bound. Illustrate with example of knapsack problem.
10.	Solve the following instance of the knapsack problem by branch & bound algorithm. n = 4, W (1:4) = {10,7,8,4}, P (1:4) = {100,63,56,12}, knapsack capacity M = 16
Sol	The solution to the knapsack problem with the given parameters (n=4, weights={10, 7, 8, 4}, profits={100, 63, 56, 12}, knapsack capacity M=16) using the Least Cost Branch and Bound algorithm is as follows: Best profit: 124 Best solution (1 for selected items, 0 for others): [1, 0, 1, 1] Explanation:

QB for Design and Analysis of Algorithm

	The best profit achieved is 124. The best solution indicates that items 1, 3, and 4 are selected (items are 0-indexed), which have weights 10, 8, and 4 respectively, and profits 100, 56, and 12 respectively. These items fit within the knapsack's capacity of 16.
11.	Explain branch & bound technique and different strategies used in it like LCBB, FIFOBB, compare LCBB & FIFOBB.
12.	Consider the travelling salesman instance defined by cost matrix. [18] $\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \end{bmatrix}$ 16 4 7 16 ∞ a) Obtain reduced matrix. b) Obtain the portion of state space tree generated by LCBB.
1211	Solve Travelling Salesman Problem using Branch and Bound Algorithm using following cost matrix

∞	10	15	20
5	∞	9	10
6	13	∞	12
8	8	9	∞

QB for Design and Analysis of Algorithm

14	Given items I= (I1, I2, I3, I4), Weight w=(2,4,6,9) and Profit p=(10,10,12,18). Let us consider the capacity of the knapsack W = 15. Find the maximum profit using LC Branch and Bound Method

Travelling Salesman Problem \| Branch & Bound

Design & Analysis of Algorithms
Spread the love

Travelling Salesman Problem-

You are given-

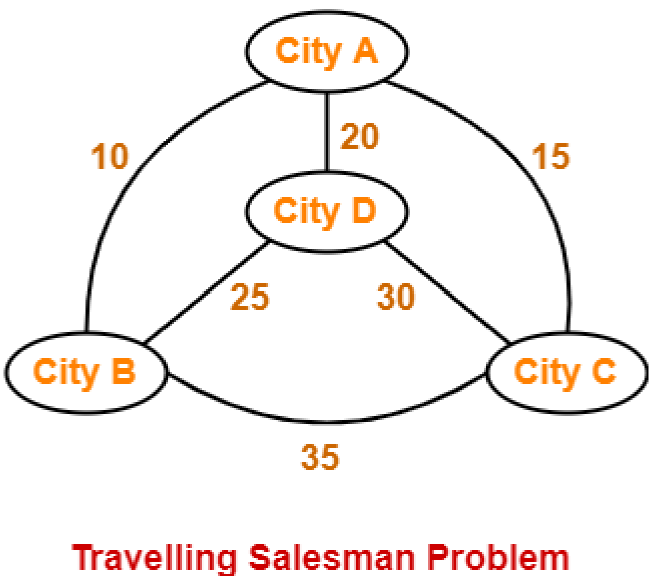
- A set of some cities
- Distance between every pair of cities

Travelling Salesman Problem states-

- A salesman has to visit every city exactly once.
- He has to come back to the city from where he starts his journey.
- What is the shortest possible route that the salesman must follow to complete his tour?

Example-

QB for Design and Analysis of Algorithm

	The following graph shows a set of cities and distance between every pair of cities-  If salesman starting city is A, then a TSP tour in the graph is- **A → B → D → C → A** Cost of the tour = 10 + 25 + 30 + 15 = **80 units** In this article, we will discuss how to solve travelling salesman problem using branch and bound approach with example.

	PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043
	Department of Information Technology **S.No.-27, Pune Satara Road, Dhankawadi, Pune-411043**

QB for Design and Analysis of Algorithm

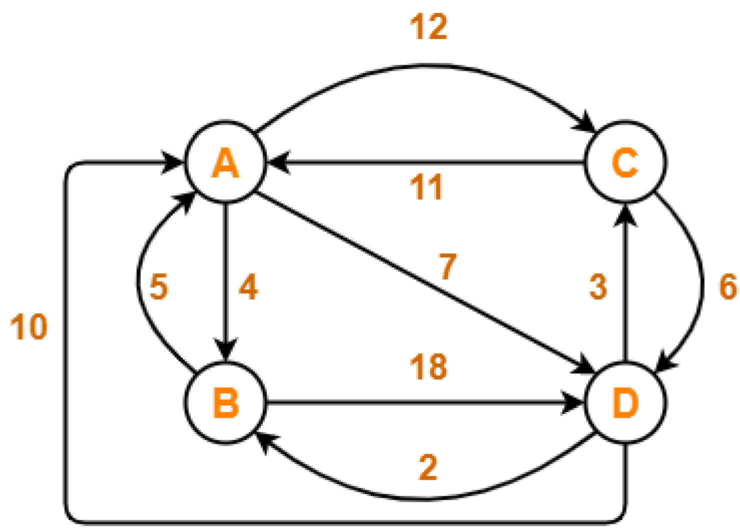
PRACTICE PROBLEM BASED ON TRAVELLING SALESMAN PROBLEM USING BRANCH AND BOUND APPROACH-

Problem-

Solve Travelling Salesman Problem using Branch and Bound Algorithm in the following graph-

1. Now Playing

QB for Design and Analysis of Algorithm



Solution-

Step-01:

Write the initial cost matrix and reduce it-

QB for Design and Analysis of Algorithm

$$
\begin{array}{c}
 \\
\begin{array}{c} A \\ B \\ C \\ D \end{array}
\end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix}
\infty & 4 & 12 & 7 \\
5 & \infty & \infty & 18 \\
11 & \infty & \infty & 6 \\
10 & 2 & 3 & \infty
\end{bmatrix}
\end{array}
$$

Rules

- To reduce a matrix, perform the row reduction and column reduction of the matrix separately.
- A row or a column is said to be reduced if it contains at least one entry '0' in it.

Row Reduction-

Consider the rows of above matrix one by one.

If the row already contains an entry '0', then-

- There is no need to reduce that row.

If the row does not contains an entry '0', then-

QB for Design and Analysis of Algorithm

- Reduce that particular row.
- Select the least value element from that row.
- Subtract that element from each element of that row.
- This will create an entry '0' in that row, thus reducing that row.

Following this, we have-

- Reduce the elements of row-1 by 4.
- Reduce the elements of row-2 by 5.
- Reduce the elements of row-3 by 6.
- Reduce the elements of row-4 by 2.

Performing this, we obtain the following row-reduced matrix-

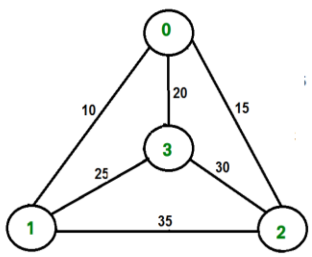
$$\begin{array}{c c} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} \infty & 0 & 8 & 3 \\ 0 & \infty & \infty & 13 \\ 5 & \infty & \infty & 0 \\ 8 & 0 & 1 & \infty \end{bmatrix} \end{array}$$

<u>Column Reduction-</u>

QB for Design and Analysis of Algorithm

	Consider the columns of above row-reduced matrix one by one. If the column already contains an entry ‘0’, then- • There is no need to reduce that column. If the column does not contains an entry ‘0’, then- • Reduce that particular column. • Select the least value element from that column. • Subtract that element from each element of that column. • This will create an entry ‘0’ in that column, thus reducing that column. Following this, we have- • There is no need to reduce column-1. • There is no need to reduce column-2. • Reduce the elements of column-3 by 1. • There is no need to reduce column-4. Performing this, we obtain the following column-reduced matrix-

QB for Design and Analysis of Algorithm



PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043

Department of Information Technology

S.No.-27, Pune Satara Road, Dhankawadi, Pune-411043

QB for Design and Analysis of Algorithm

Unit-VI

1.	Define P, NP, NP-Complete, NP-hard Classes
2.	Explain complexity classes P and NP. And differentiate between NP complete and NP Hard.
3.	Prove that Clique Decision problem is NP complete
4.	Differentiate between : i) P class and NP Class. ii) NP complete and NP Hard.
5.	Prove that Satisfiability problem in NP complete.
6.	Explain Nondeterministic algorithm? Write the Nondeterministic algorithm for searching the element of an array.
7.	Show that 3-SAT problem is NP-Complete.
8.	Explain NP-Hard, NP-Complete, Decision problem & Polynomial time algorithm
9.	What is a Non-deterministic algorithm? Write the Nondeterministic algorithm for sorting the element of an array.

10.	Is the vertex cover problem NP-hard?
11.	What are steps to prove NP-completeness of a problem? Prove that vertex cover problem is NP-complete
12	What is Boolean Satisfiability Problem? Explain 3-SAT problem. Prove 3-SAT in NP-complete.
13	What is deterministic and non-deterministic algorithm? Explain with example
14	What do you mean by NP Hard and NP Complete problems? Give an Example.
15	Prove that Satisfiability problem in NP complete
16	Explain Nondeterministic algorithm? Write the Nondeterministic algorithm for searching the element of an array
17	Write a non-deterministic algorithm for searching key elements and explain that function