

**Department of Information Technology**

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

**wwwQuestion Bank****Subject Name: Design and Analysis of Algorithm****UNIT –I ( 2 Marks Questions)****1. Define Space complexity and time Complexity?**

The space complexity of an algorithm is the amount of memory it needs to run to completion. The time complexity of an algorithm is the amount of computer time it needs to run to completion.

**2. What is Algorithm specification?****3. Define big-oh and Big-Omega notation?**

A function  $t(n)$  is said to be in  $O(g(n))$ , denoted  $t(n) O(g(n))$ , if  $t(n)$  is bounded above by some constant multiple of  $g(n)$  for all large  $n$ , i.e., if there exist some positive constant  $c$  and some nonnegative integers  $n_0$  such that

$$t(n) \leq cg(n) \text{ for all } n \geq n_0$$

**4. State the weighing, collapse rules in sets?**

Weighted Union. A low-cost approach to reducing the height is to be smart about how two trees are joined together. One simple technique, called the weighted union rule, joins the tree with fewer nodes to the tree with more nodes by making the smaller tree's root point to the root of the bigger tree.

The main idea of collapsing find is: if the immediate parent of an element isn't the head of the subset, then find the subset and collapse the element by setting its immediate parent to the head of the subset

**5. What is asymptotic notation?**

Asymptotic notations are used to represent the complexities of algorithms for asymptotic analysis. These notations are mathematical tools to represent the complexities. There are three notations that are commonly used.

**6. What are UNION, FIND operations?**

Union Operation: Merge a smaller set to a larger set if two elements are disjoint.

Find Operation: Find root parent and determine if two elements are in the same set.

**7. Define Heap?**

The heap is one maximally efficient implementation of an abstract data type called a priority queue, and in fact, priority queues are often referred to as "heaps", regardless of how they may be implemented. In a heap, the highest (or lowest) priority element is always stored at the root.

**8. Explain the UNION algorithm using weighting rule?**

Weighted Union. A low-cost approach to reducing the height is to be smart about how two trees are joined together. One simple technique, called the weighted union rule, joins the tree with fewer nodes to the tree with more nodes by making the smaller tree's root point to the root of the bigger tree.

**9. What is an algorithm?**

An algorithm is a sequence of unambiguous instructions for solving a problem. i.e., for obtaining a required output for any legitimate input in a finite amount of time

**10. What do you mean by Amortized Analysis?**

Amortized analysis finds the average running time per operation over a worst case sequence of operations. Amortized analysis differs from average-case performance in that probability is not involved; amortized analysis guarantees the time per operation over worst-case performance.

**11. What are algorithm design techniques?**

Algorithm design techniques ( or strategies or paradigms) are general approaches to solving problems algorithmically, applicable to a variety of problems from different areas of computing. General design techniques are:



## Department of Information Technology

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

- (i) Brute force (ii) divide and conquer
- (iii) decrease and conquer (iv) transform and conquer
- (v) greedy technique (vi) dynamic programming
- (vii) backtracking (viii) branch and bound

### 12. How is an algorithm's time efficiency measured?

Time efficiency indicates how fast the algorithm runs. An algorithm's time efficiency is measured as a function of its input size by counting the number of times its basic operation (running time) is executed. Basic operation is the most time consuming operation in the algorithm's innermost loop.

### 13. Give the two major phases of performance evaluation

Performance evaluation can be loosely divided into two major phases:

- (i) a prior estimates (performance analysis)
- (ii) a Posterior testing(performance measurement)

### 14. Define order of an algorithm

Measuring the performance of an algorithm in relation with the input size  $n$  is known as order of growth.

### 15. How is the efficiency of the algorithm defined?

The efficiency of an algorithm is defined with the components.

- (i) Time efficiency -indicates how fast the algorithm runs
- (ii) Space efficiency -indicates how much extra memory the algorithm needs

### 16. What are the characteristics of an algorithm?

Every algorithm should have the following five characteristics

- (i) Input
- (ii) Output
- (iii) Definiteness
- (iv) Effectiveness
- (v) Termination

Therefore, an algorithm can be defined as a sequence of definite and effective instructions, which terminates with the production of correct output from the given input. In other words, viewed little more formally, an algorithm is a step by step formalization of a mapping function to map input set onto an output set.

### 17. What do you mean by time complexity and space complexity of an algorithm?

Time complexity indicates how fast the algorithm runs. Space complexity deals with extra memory it require. Time efficiency is analyzed by determining the number of repetitions of the basic operation as a function of input size. Basic operation: the operation that contributes most towards the running time of the algorithm The running time of an algorithm is the function defined by the number of steps (or amount of memory) required to solve input instances of size  $n$ .

### 18. Define Big Omega Notations

A function  $t(n)$  is said to be in  $\Omega(g(n))$ , denoted  $t(n) \in \Omega(g(n))$ , if  $t(n)$  is bounded below by some positive constant multiple of  $g(n)$  for all large  $n$ , i.e., if there exist some positive constant  $c$  and some nonnegative integer  $n_0$  such that  $t(n) \geq cg(n)$  for all  $n \geq n_0$

### 19. What are the different criteria used to improve the effectiveness of algorithm?

- (i) The effectiveness of algorithm is improved, when the design, satisfies the following constraints to be minimum.

**Department of Information Technology**

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

Time efficiency - how fast an algorithm in question runs.

Space efficiency – an extra space the algorithm requires

(ii) The algorithm has to provide result for all valid inputs.

**20. Analyze the time complexity of the following segment:**

for(i=0;i&lt;N;i++)

for(j=N/2;j&gt;0;j--)

sum++;

Time Complexity=  $N * N/2$ =  $N^2 / 2$  $\in O(N^2)$ **21. Define Big Theta Notations**

A function  $t(n)$  is said to be in  $\Theta(g(n))$ , denoted  $t(n) \in \Theta(g(n))$ , if  $t(n)$  is bounded both above and below by some positive constant multiple of  $g(n)$  for all large  $n$ , i.e., if there exist some positive constants  $c_1$  and  $c_2$  and some nonnegative integer  $n_0$  such that  $c_1 g(n) \leq t(n) \leq c_2 g(n)$  for all  $n \geq n_0$

**22. What is performance measurement?**

Performance measurement is concerned with obtaining the space and the time requirements of a particular algorithm.

**23. What is an algorithm?**

An algorithm is a finite set of instructions that, if followed, accomplishes a particular task. In addition, all algorithms must satisfy the following criteria:

- 1) input
- 2) Output
- 3) Definiteness
- 4) Finiteness
- 5) Effectiveness.

**UNIT –I ( Essay Questions)**

1. Explain the properties of an algorithm with an example.
2. Define an algorithm and write the characteristics of the algorithm.
3. Explain the process of designing an algorithm. Give characteristics of an algorithm.
4. List out the steps that need to design v an algorithm?
5. Distinguish between Algorithm and Pseudocode
6. Define time complexity and space complexity. Write an algorithm for adding  $n$  natural numbers and find the time required by that algorithm?
7. Define time and space co okmplexity. Explain with examples.
8. Describe performance analysis, space complexity and time complexity.
9. Differentiate between Bigoh and Omega notation with example.
10. Differentiate between Best, average and worst-case efficiency?
11. Discuss the concepts of asymptotic notations and its properties?
12. Explain asymptotic notations.
13. Explain about Disjoint set operations.



## Department of Information Technology

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

14. Discuss union and find algorithms in detail with an example.

### UNIT –II( 2 MARKS Questions)

#### 1. Define the divide and conquer method or Discuss divide and conquer method?

Given a function to compute on 'n' inputs the divide-and-conquer strategy suggests splitting the inputs into 'k' distinct subsets,  $1 < k < n$ , yielding 'k' subproblems. The subproblems must be solved, and then a method must be found to combine sub-solutions into a solution of the whole. If the subproblems are still relatively large, then the divide-and-conquer strategy can possibly be reapplied.

#### 2. Define control abstraction of DAC

A control abstraction we mean a procedure whose flow of control is clear but whose primary operations are by other procedures whose precise meanings are left undefined.

Write the Control abstraction for Divide-and-conquer

...pk,  $k \geq 1$ ;

Apply DAndC to each of these subproblems

Return combine (DAnd C(p1) DAnd C(p2),---, DAnd (pk));

}

}

#### 3. Write Analysis for the quick sort?

#### 4. Define Binary search?

#### 5. What is the substitution method?

One of the methods for solving any such recurrence relation is called the substitution method.

#### 6. What is the binary search?

If 'q' is always chosen such that 'aq' is the middle element (that is,  $q = \lfloor (n+1)/2 \rfloor$ ), then the resulting search algorithm is known as binary search.

#### 7. Give computing time for Binary search?

In conclusion we are now able completely describe the computing time of binary search by giving formulas that describe the best, average and worst cases.

Successful searches

$\Theta(1)$   $\Theta(\log n)$   $\Theta(\log n)$

best average worst

Unsuccessful searches

$\Theta(\log n)$

best, average, worst

#### 8. What is the maximum and minimum problem?

The problem is to find the maximum and minimum items in a set of 'n' elements. Though this problem may look so simple as to be contrived, it allows us to demonstrate divide-and-conquer in simple setting.

**Department of Information Technology**

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

**9. What is the Quick sort?**

Quicksort is divide and conquer strategy that works by partitioning its input elements according to their value relative to some preselected element(pivot). it uses recursion and the method is also called partition –exchange sort.

**10. Write the Analysis for the Quick sort.**

$O(n \log n)$  in average and best cases

$O(n^2)$  in worst case

**11. what is Merge sort? and Is insertion sort better than the merge sort?**

Merge sort is divide and conquer strategy that works by dividing an input array in to two halves, sorting them recursively and then merging the two sorted halves to get the original array sorted

Insertion sort works exceedingly fast on arrays of less than 16 elements, though for large 'n' its computing time is  $O(n^2)$ .

**12. Write a algorithm for straightforward maximum and minimum?**

Algorithm straight MaxMin(a,n,max,min)

//set max to the maximum and min to the minimum of a[1:n]

```
{
max := min: = a[i];
for i = 2 to n do
{
if(a[i] > max) then max: = a[i];
if(a[i] > min) then min: = a[i];
}
}
```

**13.What is general divide and conquer recurrence?**

Time efficiency  $T(n)$  of many divide and conquer algorithms satisfies the equation  $T(n) = a.T(n/b) + f(n)$ . This is the general recurrence relation.

**14.. Write the algorithm for Iterative binary search?**

Algorithm BinSearch(a,n,x)

//Given an array a[1:n] of elements in nondecreasing

// order,  $n > 0$ , determine whether x is present

```
{
low := 1;
high := n;
while (low < high) do
{
mid := [(low+high)/2];
if(x < a[mid]) then high:= mid-1;
else if (x > a[mid]) then low:=mid + 1;
else return mid;
}
```



## Department of Information Technology

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

```
}
return 0;
}
```

### 15. Describe the recurrence relation for merge sort?

If the time for the merging operation is proportional to  $n$ , then the computing time of merge sort is described by the recurrence relation

$T(n) = a n = 1$ ,  $a$  a constant

$2T(n/2) + n$ ,  $n > 1$ ,  $c$  a constant

## GREEDY METHOD

1. What is the Control abstraction of greedy method?
2. Define Minimum cost spanning tree? What are the methods to find MST?
3. List out any three problems addressed by Greedy methods?
4. Define

- i) Principles of optimality
- ii) Feasible solution
- iii) Optimal solution.

### 5. Distinguish between Prim's and Kruskal's spanning tree algorithm

### 6. Explain the greedy method.

Greedy method is the most important design technique, which makes a choice that looks best at that moment. A given 'n' inputs are required us to obtain a subset that satisfies some constraints that is the feasible solution. A greedy method suggests that one can devise an algorithm that works in stages considering one input at a time.

### 7. Define feasible and optimal solution.

Given  $n$  inputs and we are required to form a subset such that it satisfies some given constraints then such a subset is called feasible solution. A feasible solution either maximizes or minimizes the given objective function is called as optimal solution

### 8. Write the control abstraction for greedy method.

Algorithm Greedy ( $a, n$ )

```
{
solution=0;
for i=1 to n do
{
x= select(a);
if feasible(solution ,x) then
solution=Union(solution ,x);
}
return solution;
}
```



**Department of Information Technology**

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

**9. What are the constraints of knapsack problem?**To maximize  $\sum p_i x_i$ 

$$1 \leq i \leq n$$

The constraint is :  $\sum w_i x_i \leq m$  and  $0 \leq x_i \leq 1$   $1 \leq i \leq n$ 

$$1 \leq i \leq n$$

where m is the bag capacity, n is the number of objects and for each object i  $w_i$  and  $p_i$  are the weight and profit of object respectively.**10. What is a minimum cost spanning tree?**

A spanning tree of a connected graph is its connected acyclic subgraph that contains all vertices of a graph. A minimum spanning tree of a weighted connected graph is its spanning tree of the smallest weight where bweight of the tree is the sum of weights on all its edges.

A minimum spanning subtree of a weighted graph  $(G, w)$  is a spanning subtree of G of minimum weight  $w(T) = \sum w(e)$

$$e \in T$$

Minimum Spanning Subtree Problem: Given a weighted connected undirected graph  $(G, w)$ , find a minimum spanning subtree

**11. Specify the algorithms used for constructing Minimum cost spanning tree.****a) Prim's Algorithm****b) Kruskal's Algorithm****12. State single source shortest path algorithm (Dijkstra's algorithm).**

For a given vertex called the source in a weighted connected graph, find shortest paths to all its other vertices. Dijkstra's algorithm applies to graph with non-negative weights only.

**13. What is Knapsack problem?**

A bag or sack is given capacity and n objects are given. Each object has weight  $w_i$  and

profit  $p_i$ . Fraction of object is considered as  $x_i$  (i.e)  $0 \leq x_i \leq 1$ . If fraction is 1 then entire object is put into sack. When we place this fraction into the sack we get  $w_i x_i$  and  $p_i x_i$ .

**14. Write any two characteristics of Greedy Algorithm?**

\* To solve a problem in an optimal way construct the solution from given set of candidates.

\* As the algorithm proceeds, two other sets get accumulated among this one set contains

the candidates that have been already considered and chosen while the other set contains

the candidates that have been considered but rejected.

**15. What is the Greedy approach?**

The method suggests constructing solution through sequence of steps, each expanding partially constructed solution obtained so far, until a complete solution is



## Department of Information Technology

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

reached. On each step, the choice must be

- Feasible (satisfy problem constraints)
- Locally optimal (best local choice among all feasible choices available on that step)
- Irrevocable (once made, it can't be changed)

### 16. What are the steps required to develop a greedy algorithm?

- \* Determine the optimal substructure of the problem.
- \* Develop a recursive solution.
- \* Prove that at any stage of recursion one of the optimal choices is greedy choice. Thus it is always safe to make greedy choice.
- \* Show that all but one of the sub problems induced by having made the greedy choice are empty.
- \* Develop a recursive algorithm and convert into iterative algorithm.

17. Write the difference between the Greedy method and Dynamic programming.

Greedy method      Dynamic programming

Only one sequence of decision is generated.

Many number of decisions are generated.

It does not guarantee to give an optimal solution always.

It definitely gives an optimal solution always.

18. State the requirement in optimal storage problem in tapes.

Finding a permutation for the  $n$  programs so that when they are stored on the tape in this order the MRT is minimized. This problem fits the ordering paradigm.

16. State efficiency of Prim's algorithm.

$O(|V|^2)$  (WEIGHT MATRIX AND PRIORITY QUEUE AS UNORDERED ARRAY)

$O(|E| \log |V|)$  (ADJACENCY LIST AND PRIORITY QUEUE AS MIN-HEAP)

19. State Kruskal Algorithm.

The algorithm looks at a MST for a weighted connected graph as an acyclic subgraph with  $|V|-1$  edges for which the sum of edge weights is the smallest.

20. State efficiency of Dijkstra's algorithm.

$O(|V|^2)$  (WEIGHT MATRIX AND PRIORITY QUEUE AS UNORDERED ARRAY)

$O(|E| \log |V|)$  (ADJACENCY LIST AND PRIORITY QUEUE AS MIN-HEAP)

21. Differentiate subset paradigm and ordering paradigm

subset paradigm

At each stage a decision is made regarding whether a particular input is in an optimal solution (generating sub optimal solutions)

ordering paradigm

For problems that do not call for selection of optimal subset, in the greedy manner we make decisions by considering inputs in some order. Example: knapsack, MST, Optimal storage on tapes





## Department of Information Technology

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

### UNIT –II( Essay Questions)

#### Divide and Conquer

1. Write Short notes on divide and conquer?
2. a) Write the General method and Control Abstraction of Divide – and – Conquer.  
b) What are the advantages and disadvantages of Divide – And – Conquer
3. Write Divide – And – Conquer recursive Merge sort algorithm and derive the time. complexity of this algorithm?
4. a) Distinguish between Merge sort and quick sort.  
b) Explain Recursive Binary search algorithm with suitable examples.  
c) Discuss the time complexity of Binary search algorithm for best and worst case.
5. Define Merge sort with example.
- 6) a) Explain Merge sort (Stable sort (or) Not in-place) algorithm and trace this algorithm for n=8 elements: 24,12, 35, 23,45,34,20,48  
b) Derive the time complexity of Merge sort algorithm for all cases.
7. Describe Quick Sort with suitable example?
8. a) Explain partition exchange sort (Quick Sort (or) in-place Sort (or) unstable sort) algorithm and trace this algorithm for n=8 elements: 24,12, 35, 23,45,34,20,48.6  
b) Write Divide – And – Conquer recursive Quick sort algorithm and analyze the algorithm for average time complexity. c) Derive the time complexity of Quick sort algorithm for worst case.
9. Illustrate Merge sort algorithm and discuss its time complexity.
10. Describe merge sort algorithm and explain with an example?
11. Sort the keys using merge sort (100,300,150,450,250,350,200,400,500)
12. Simulate Quick sort algorithm for the following example 25,36,12,4,5,16,58,54,24,16,9,65,78
13. Give the general procedure of divide and conquer method.
14. Write about quick sort method with example.
15. Illustrate the general method of divide and conquer technique.
16. Write an algorithm for Binary search and discuss its complexity.
17. Write binary search algorithm and analyse its time complexity? What is greedy method. List out Applications?

#### Greedy method

1. Write the general method and Control Abstraction of Greedy method
2. a) State the Job – Sequencing Deadline Problem

**Department of Information Technology**

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

- b) Find an optimal sequence to the  $n=5$  Jobs where profits  $(P_1, P_2, P_3, P_4, P_5) = (20, 15, 10, 5, 1)$  and deadlines  $(d_1, d_2, d_3, d_4, d_5) = (2, 2, 1, 3, 3)$ .
3. Derive time complexity of job sequencing with deadlines. Obtain the optimal solution when  $n=5$ ,  $(p_1, p_2, \dots) = (20, 15, 10, 5, 1)$  and  $(d_1, d_2, \dots) = (2, 2, 1, 3, 3)$ .
3. What is a Spanning tree? Explain Prim's Minimum cost spanning tree algorithm with suitable example and also find the time complexity.
- b) What is a Minimum Cost Spanning tree? Explain Kruskal's Minimum cost spanning tree algorithm with suitable example and also find the time complexity.
4. Find the minimum cost spanning tree for the graph given below?
- 5a) Discuss the single – source shortest paths (i.e. Dijkstra's) algorithm with suitable example and also find the time complexity.
6. Write Greedy algorithm for sequencing unit time jobs with deadlines and profits?
7. Write kruskals algorithm and explain with an example to find minimum spanning tree?
8. a) State the Greedy Knapsack? Write the algorithm for Greedy knapsack and also compute the time complexity.
- b) Find an optimal solution to the knapsack instance  $n=7$  objects and the capacity of knapsack  $m=15$ . The profits and weights of the objects are  $(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (10, 5, 15, 7, 6, 18, 3)$   $(W_1, W_2, W_3, W_4, W_5, W_6, W_7) = (2, 3, 5, 7, 1, 4, 1)$ .
9. Consider the following knapsack instance problem  $n=3$ ,  $m=20$ ,  $(P_1-P_3) = (25, 24, 15)$  and  $(W_1-W_3) = (18, 15, 10)$   
Solve by Greedy approach?
10. Consider the following knapsack instance problem  $n=4$ ,  $m=15$   $(P_1-P_4) = (10, 10, 12, 18)$  and  $(W_1-W_4) = (2, 4, 6, 9)$   
Solve by Greedy approach?
11. Consider the following knapsack instance problem  $n=5$ ,  $m=12$   $(P_1-P_5) = (10, 15, 6, 8, 4)$  and  $(W_1-W_5) = (4, 6, 3, 4, 2)$  Solve by Greedy approach?
12. Consider the following knapsack instance problem  $n=7$ ,  $m=15$ ,  $(P_1-P_7) = (10, 5, 15, 7, 6, 18, 3)$  and  $(W_1-W_7) = (2, 3, 5, 7, 1, 4, 1)$  Solve by Greedy approach?

**UNIT-III****Dynamic Programming**



## Department of Information Technology

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

### 1. Write the difference between the Greedy method and Dynamic programming.

Greedy method      Dynamic programming

1. Only one sequence of decision is generated.      1. Many numbers of decisions are generated.

2. It does not guarantee to give an optimal solution always.      2. It gives an optimal solution always.

### 2. Define dynamic programming.

Dynamic programming is an algorithm design method that can be used when a solution to the problem is viewed as the result of sequence of decisions. It is technique for solving problems with overlapping subproblems.

### 3. What are the features of dynamic programming?

- Optimal solutions to sub problems are retained to avoid recomputing of their values.
- Decision sequences containing sub sequences that are sub optimal are not considered.
- It definitely gives the optimal solution always.

### 4. What are the drawbacks of dynamic programming?

- Time and space requirements are high, since storage is needed for all level.
- Optimality should be checked at all levels.

### 5. Write the general procedure of dynamic programming.

The development of dynamic programming algorithm can be broken into a sequence of 4 steps.

1. Characterize the structure of an optimal solution.
2. Recursively define the value of the optimal solution.
3. Compute the value of an optimal solution in the bottom-up fashion.
4. Construct an optimal solution from the computed information.

### 6. Define principle of optimality.

It states that an optimal solution to any of its instances must be made up of optimal solutions to its sub instances.

### 7. Define multistage graph

A multistage graph  $G=(V,E)$  is a directed graph in which the vertices are partitioned into  $K \geq 2$  disjoint sets  $V_i, 1 \leq i \leq k$ . The multi stage graph problem is to find a minimum cost paths from  $s$ (source) to  $t$ (sink).

### 8. Define All pair shortest path problem

Given a weighted connected graph, all pair shortest path problem asks to find the lengths of shortest paths from each vertex to all other vertices.

9. State the time efficiency of Floyd's algorithm

$O(n^3)$  It is cubic

10. Define OBST

☐ Dynamic programming. Used

☐ If probabilities of searching for elements of a set are known then finding optimal BST for which the average number of comparisons in a search is smallest possible.

11. Define Catalan number

The total number of binary search trees with  $n$  keys is equal to  $n$ th Catalan number

$C(n) = \frac{(2n)!}{(n+1)!n!}$  for  $n \geq 0, C(0)=1$

**Department of Information Technology**

email: hodit@matrusri.edu.in

Website: www.matrusri.edu.in

12. State time and space efficiency of OBST

Space Efficiency: Quadratic

Time Efficiency : Cubic.

13.Explain dominance rules in 0/1 Knapsack problem.

1.Design a three-stage system with devices D1, D2, D3. The costs are \$40, \$30, \$20

respectively. The cost of the system is to be no more than \$110 the

reliability of each device type is 0.9, 0.7, 0.5.

2.State the principle of optimality. Find two problems for which the principle does not hold

3.Briefly explain Multistage graphs with suitable examples.

4.Explain reliability design problem with an example?

5. Write a short note on any two of the following.

Optimal Binary Search Tree

6. a) Solve the following 0/1 Knapsack problem using dynamic programming  $P = (11, 21, 31, 33)$ , $W = (2, 11, 22, 15)$ ,  $C=40$ ,  $n=4$ .b) Consider three stages of a system with  $r_1=0.3$ ,  $r_2=0.5$ ,  $r_3=0.2$  and  $c_1=30$ ,  $c_2=20$ ,  $c_3=30$ Where the total cost of the system is  $C=80$  and  $u_1=2$ ,  $u_2=3$ ,  $u_3=2$  find the reliability design.

7) (a) Define merging and purging rules in 0/1 Knapsack problem.

(b) Write an algorithm for all pairs shortest path. Explain with an example

8) Consider 4 elements  $a_1 < a_2 < a_3 < a_4$  with  $q_0=0.25$ ,  $q_1=3/16$ ,  $q_2=q_3=q_4=1/16$ .  $p_1=1/4$ ,  $p_2=1/8$ ,  $p_3=p_4=1/16$ .

(i) Construct the optimal binary search tree as a minimal cost tree.

(ii) Construct the table of values  $W_{ij}$ ,  $C_{ij}$ ,  $V_{ij}$  computed by the algorithm to compute the roots of optimal subtrees

9) (a) Dynamic Programming vs. Divide and Conquer

#  
(ISO



# Matrusri Engineering College

(Sponsored by Matrusri Education Society, Estd.1980)  
(Approved by AICTE & Affiliated to Osmania University)  
16-1-486, Saidabad, Hyderabad-500059. Ph: 040-24072764  
9001:2015 Certified)



## Department of Information Technology

email: [hodit@matrusri.edu.in](mailto:hodit@matrusri.edu.in)

Website: [www.matrusri.edu.in](http://www.matrusri.edu.in)

(b) Dynamic Programming vs. Greedy method

10) (a) Find the shortest tour of TSP for the following graph using dynamic Programming

(b) What is the best method between greedy method and dynamic programming to solve single

source shortest path problem? Justify your answer with example.

11. Let  $n=4$  and  $(a_1, a_2, a_3, a_4)$  Construct optimal binary search for  $(a_1, a_2, a_3, a_4) = (\text{do}, \text{if}, \text{int}, \text{while})$ ,  $p(1 : 4) = (3, 3, 1, 1)$   $q(0 : 4) = (2, 3, 1, 1, 1)$