16.Question Bank

UNIT I

Short Questions:

1. If $f(n)=a_m n^m+\ldots..a_1n+a_o$.then prove that $f(n)=O(n^m)$
2. Establish the relationship between Big-oh and Omega
3. What do you mean by Algorithm?
4. Define Big-oh notation?
5. What do you mean by Divide and Conquer Strategy?
6. What are the properties of Big-oh notations?
7. What is called substitution Method?
8. Differentiate Time complexity from Space Complexity?
9. What is Recurrence Relation?
10. What do you mean by Amortized Analysis?
11. How is the efficiency of the algorithm defined?
12. How is an algorithm's time efficiency measured?
13. Define direct recursive and indirect recursive algorithms?
14. What are the characteristics of an algorithm?
15. Analyze the time complexity of the following segment:

```
for(i=0;i<N;i++)
for(j=N/2;j>0;j--)
sum++;
```

Long Questions:

1. Prove that every polynomial $p(n)=a_kn^k+a_{k-1}n^{k-1}+$_ _ _ _ _ =a0 with $a_k>0$ belongs to $\theta(n^k)$

2. Prove that exponential functions a^n have different order of growth for different values of base a>0

3. Set up and solve a recurrence relation for the number of calls made by F(n), the recursive algorithm for computing n!

4. Assume that T satisfies $T(n)\leq a^T([n/b])+f(n)$, $f(n)=O(n^k)$, and $a<b^k$. Choose a suitable constant C and then use induction to prove that $T(n)\ )\leq cn^k\log n+T(1)+T(2)+$ _ _ _ _T([b]) for all $n\geq1$. Prove that $T(n)=O(n^k\log n)$.

5. With an example problem derive amortized complexity of an algorithm?

6 Show the complexity of Fibonacci function, F(n) is $\Omega((3/2)^n)$

7. Give a recursive algorithm to compu7te the produce of two positive integers m and n using only addition.

8. Develop an algorithm which converts a Roman numeral into an Arabic integer.

10. What is an algorithm? What are the properties of an algorithm? Explain with example?

11. What are the rules that are to be followed when writing an algorithm. Explain with an example?

12. What is performance analysis of an algorithm? Explain various notations used?

13. When a mortised analysis use used to be measure the performance of an algorithm? Explain in detail?

14. What happens to the worst case runtime of quick sort if the median of the given key is used as spitter key? Derive the relation?

15. Input is an array of numbers where each number is an integer in the range [O,N} (for some N>>n) present algorithm that runs in the worst case in time O(n(logn/logn)) and check whether these n numbers are distinct, and the algorithm should use only O9n) space? Design.

16. The sets A and B have m and n elements from a linear order. These sets are not necessarily sorted. Also assume mεn. Show how to compute $A \cup B$ and $A \cap B$ in O(nlogn) time.

17. Give a proof which shows that the recurrence relation $T(n)=mT(n/2)+an^2$ is satisfied by T(n)=O(nlogn).

18. Show how quick sort procedure sorts the following sets of keys (1,1,1,1,1,1,1) and (5,5,8,3,4,3,2).

19. Write an algorithm for quick sort and derive a recurrence relation to show the performance?

20. What is worst case complexity of merge sort? Derive the recurrence relation with help of an algorithm?

21. Give recursive and non recursive algorithms to binary search? Give their performances?

22. Give the control abstraction of divide and conquer strategy with an example?

23 What is the advantage of Strassens matrix multiplication? Explain?

24. Compare the quicksort and merge sort algorithms?

25. Compute 2101*1130 by applying the divide and conquer algorithm

UNIT II

Short Questions:

1. What is collapsing rule?
2. What are AND/OR graphs?
3. Explain about Game trees?
4. Write the non recursive algorithms for post order traversals?
5. What is connected and Biconnected components?
6. Explain about articulation point?
7. What is spanning tree?Explain with examples?
8. Write the algorithm for Breadth first search?
9. What is weighting rule for union?
10. Write the algorithm for union and find?

Long Questions:

1. Experimentally compare the performance of simple union and simple find with weighted union and collapsing find and generate a random sequence of union and find operations?

2. What that any connected = undirected graph G = (V_2,E) satisfy |E|≥|V|-|

3 Shoat that it a directed graph or undirected graph contain path between two vertices u and v, then it contains a simple path between u and V, show that if a directed graph contains a cycle, then it contains a simple cycle.

4. Suppose if a set of n elements contains distinct elements show that at most n-1 unions can be performed before the number of sets becomes 1.

5. With an example write the algorithm for simple union and measure the algorithms performance?

6. With an example write the algorithm for simple find and give the algorithms performance?

7. Bring out the differences between simple and weighted union? Explain with example?

8. Give the differences between simple find and collapsing Find? Give a suitable example?

10. What are connected components explain with an example.

11. What is meant by biconnected components, with example give how it is different from components.

12. What is degenerate tree? Explain with an example?

13. Show that is u unions are performed then atleast max{n-2u, 0} singleton sets remain?

14. Explain about AND/OR graphs with an example?

15. What are Game trees explain for an Tic-Tac-Toe problem?

UNIT III

Short Questions:

1. Explain the greedy method.
2. Define feasible and optimal solution.
3. Write the control abstraction for greedy method.
4. What are the constraints of knapsack problem?
5. What is a minimum cost spanning tree?
6. Specify the algorithms used for constructing Minimum cost spanning tree.
7. State single source shortest path algorithm (Dijkstra’s algorithm).
8. Write any two characteristics of Greedy Algorithm?
9. What is the Greedy approach?
10. What are the steps required to develop a greedy algorithm?

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

12. Define dynamic programming.

13. What are the features of dynamic programming?

14. Define principle of optimality.

15. Define OBST

16. Write the general procedure of dynamic programming.

17. Define multistage graph

18. Define All pair shortest path problem

19. State time and space efficiency of OBST

20. What are the drawbacks of dynamic programming?

Long Questions:

1. Write a pseudocode of the greedy algorithm for change making problem with an amount η and coin denominations $d_1 > d_2 > d_3$_ _ _ >dm as its input. What is the efficiency class of your algorithm as a function of η.

2. Is the notation of a minimum cost spanning tree is applicable to a connected graph> Do we have to check the graph's connectivity before applying the algorithms (prism's/Kruskal's) or can these algorithms does it by themselves? Justify?

3. Design a linear-time algorithm for solving the single source shortest path problem for DAG represented by their adjancency linked list?

4. By considering the complete graph with n vertices, show that the number of spanning trees in an n vertex graph can be greater than 2^{n-1}-2.

5. How minimum cost spanning tee is constructed? Explain with the help of prism's algorithm?

6. Write Kuskal's algorithm for minimum spanning tree?

7. What is 0/1 knapsack problem? Explain with an example?

8. How job sequencing is implemented by using dead links? Explain

9. Give the general method of greedy algorithms? Explain with an example?

10. What are Feasible solution, optimal solutions and the condition of optimality explain with the help of 0/1 knapsack problem?

11. If the set of job are 4 and profits are (100,20,15,27) and dead links are (2,1,2,1) the find out optimal solutions?

12. What is single source shortest path problem,? Explain with an example?

13. Write an algorithm to multiply two matrices of sizes nxm, mxp and prove number of multiplications needed is nmp?

14. Write a dynamic programming algorithm to compute a^n based on the formulas

    1. an = $a^{n/2}a^{n/2}$, n even
       an = $a^{(n/-1)2}a^{n-1/2}$, n odd

15. Define $F = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$, show that $F = \begin{bmatrix} f_{n-1} & f_n \\ f_n & f_{n+1} \end{bmatrix}$, n≥1 where f_n is the nth Fibonacci number nad f_0 is defined as 0.

16. Show that the number of different binary trees with n nodes is $\frac{1}{n+1}\binom{2n}{n}$

17. What is the need of matrix chain multiplication? How it is implemented?

18. Give the control abstraction of dynamic programming with an example?

19. Can the same problem solved by using dynamic programming and greedy method? Explain with example?

20. Show that the computing time of OBST is $O(n^2)$?

21. What is Travelling sales person problem and what are its applications.

22. Write a pseudocode of the dynamic programming algorithm for solving Optimal Binary search tree and determine its time and space efficiencies.

23. Write algorithms corresponding to ADJUST, HEAPIFY, INSERT and DELETE for the case of a min-heap represented as a complete binary tree. Explain the time complexity of HEAPIFY

24. Write the implementation of DELETE (b,s) in which an element b found at vortex v of a binary Search tree whose elements belong to Set S.

UNIT-VI

Short Questions:

1. What are the requirements that are needed for performing Backtracking?
2. Define explicit constraint and Implicit Constraints?
3. Define state space tree.
4. Define answer states.
5. Define a live node.
6. Define a E – node.
7. Define a dead node.
8. What are the factors that influence the efficiency of the backtracking algorithm?
9. Define Branch-and-Bound method.
10. What are the searching techniques that are commonly used in Branch-and-Bound

11. State 8 – Queens problem.

12. State Sum of Subsets problem.

13. State m – colorability decision problem.

14. Define chromatic number of the graph

15. What are dynamic trees?

Long Questions:

1. Design a new control abstraction of Backtracking by combining the Recursive and non-recursive control abstraction of Back tracking?

2. Design n-queue algorithm and run it for n=8,9,10 and give all the possible solutions?

3. Determine the order of magnitude of the worst case complexity of backtracking procedure which finds all Hamiltonian cycles?

4. Prove the size of the set of all subsets of n elements is $2^{n.}$

5. Let W=(5,7,10,12,15,18) and M=35, find all possible subsets of which sum to M. Draw the portion of state space tree.

6. Give solution to 4Queue problem by design and algorithm?

7. What is meant by Graphs coloring? Explain map and its planar graph with help of an algorithm?

8. Is it possible to draw Hamiltonian cycle to every graph. Explain with example.

9. What is graph coloring? Present an algorithm which finds m-coloring of a graph.

10. Write the iterative Backtracking algorithm.

11. Write an algorithm of n-queens problem.

12. Explain the depth first search algorithm for an undirected graph.

13. Design a complete LC branch-and-bound algorithm for the job sequencing with dead lines problem. Use the fixed tuple size formulation?

14. Prove the goal of the given figure is reachable from the initial state iff $\sum_{i=1}^{16} \text{LESS}(i) + X$ is even.

Figure:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

15. If T is a tour in H then T is a tour in exactly one of the graphs (V,E_i), $1 \leq i \leq r$ prove?

16. Prove that if a better function is used in LC branch-and-bound algorithm, the number of nobles generated may increase.

17. Draw the portion of the state space tree generated by LCKnap for the instance.

n=5 (P_1_ _ _ _P_5) = (10, _ _ _ 45, (w_1 _ _ _ _ 45) = (4,6,3,122) and m=12

18. Obtain the reduced cost matrix for TSP instance

$$\begin{bmatrix} \infty & 7 & 3 & 12 & 8 \\ 3 & \infty & 6 & 14 & 9 \\ 5 & 8 & \infty & 6 & 18 \\ 9 & 3 & 5 & \infty & 11 \\ 18 & 14 & 9 & 8 & \infty \end{bmatrix}$$

19. Solve the Job sequencing with dead lines? Where profits =(6,3,4,8,5) line = (2,1,2,1,1), deadlines=(3,1,4,2,14) and n=5.

20. Write a program schema DFBB?

21. Present a program schema for a FIFO Branch & Bound search for a Least-Cost answer node.

22. Explain how state space trees are used for programming nim, tic tac toe, checkers games

23. Define the term Branch & Bound and explain with an example.

24. Explain live node, E-node and dead node with an example.

UNIT-V

Short Questions:

1. What are NP- hard and Np-complete problems?
2. What is a decision problem?
3. what is approximate solution?
4. what is promising and non-promising nodes?
5. Write formula for bounding function in Knapsack problem
6. Write about traveling salesperson problem
7. Differentiate decision problem and optimization problem
8. what is class P and NP?
9. Define NP-Hard and NP-Complete problems
10. Give the time complexity and space complexity of traveling salesperson problem

Long Questions:

1. Convert the Boolean formula $B=(x_1\overline{\leftrightarrow}x_2).(\overline{x_3}+x_4x_5\overline{).}(\ x_1x_2+\ x_3x_4)$ into CNF?

2. Design Pseudocode description of the branch and bound algorithm for TSP.