2 marks:

1. State the application of Huffman's tree.

* Image Compression: reduce the amount of storage space required for digital images.
* Audio Compression: used to compress the audio files.
* Text Compression: reduce the size of text documents.
* Data Transmission: reduce the amount of band width.

2. What is Knapsack problem using greedy approach?

* The basic idea of the greedy approach is to calculate the ratio value/weight for each item and sort the item based on this ratio.
* Then take the item with the highest ratio and add it to the knapsack until we can't add the following item as a whole, and at the end, add the following item as much as possible.

3. Write the general procedure of dynamic programming?

i) Define a class of subproblems.
ii) Give a recurrence based on solving each subproblem in terms of simpler subproblems.
iii) Give an algorithm for computing the recurrence.

4. What is the formula for binomial coefficient ?

The name "binomial coefficient" comes from the participation of these numbers in the binomial formula

$$(a+b)^n = c(n,0)\,a^n + \ldots + c(n,k)\,a^{n-k}b^k + \ldots + c(n,n)\,b^n$$

where,

$$c(n,k) = c(n-1,k-1) + c(n-1,k) \text{ for } n>k>0$$

and

$$c(n,0) \text{ and } c(n,n) = 1.$$

5. What is transitive closure ?

The matrix containing the information about the existence of directed paths between any two vertices of a given graph is called transitive closure of the diagraph.

6. Define Optimal binary Search tree

An Optimal Binary Seach Tree is the one for which the average number of comparisons in a search is the smallest possible, if the probability of seaching each elements is given.

7. List out the memory functions under dynamic programming.

* Using the recurrences directly in a recursive algorithm is a top-down technique.
* It has the disadvantage that it solves common sub problem multiple times.
* This leads to poor efficiency, exponential.
* The dynamic programming technique is bottom-up, and solving all the sub-problems only once.

8. What are the applications of backtracking?

* N-Queen problem
* Hamiltonian cycle problem
* Graph colouring problems
* Subset-Sum problem
* Knapsack problem
* Constraint satisfaction problem such as crossword, verbal arithmatic etc.
* Efficient technique for parsing.

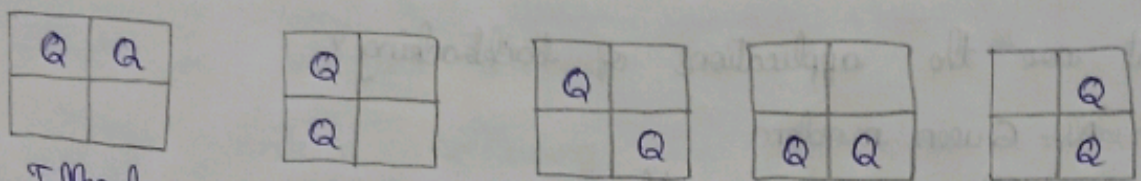
9. Differentiate explicit and implicit constraints.

Explicit constraint	Implicit constraint
* Explicit constraints are rules that restrict each x_i to take on values only from a given set. * All tuples that satisfy the explicit constraints define a possible solution space for I	* Implicit constraint are rules that determine which of the tuples in the solution space of I satisfy the criterion function. * Use a tree organizations for solution space.

10. Why 2 queens problem is not solvable? Justify your answer.

The N-Queen problem states as consider a $n \times n$ chessboard on which we have to place n queens so that no two queens attack each other by being in the same row or in the same column or on the same diagonal.

2-Queen's problem is not solvable because 2-Queens can be placed on 2×2 chess board as follows.



11. What is the principle ~~based~~ behind branch and bound technique?

Branch and bound is a method for solving optimization problems by breaking them down into smaller sub-problems and using a bounding function to eliminate sub-problems that cannot contain the optimal solution.

12. Define P and NP Problem.

P - problems

An algorithm is called polynomial time algorithm (P - class) which solves the problem in polynomial time.

Eg: searching key element

NP - problems.

* It stands for non - deterministic polynomial time.
* That means these are the kind of problems that can be sloved in non - deterministic polynomial tree.

Eg: Travelling salesman problem.

13. What is the purpose of Huffman's tree?

* The Huffman tree is the binary tree with minimum external path weight, ie, the one with the minimum sum of weighted path lengths for the given set of leaves.
* So, the goal is to build a tree with the minimum external path weight.

தமிழ் இல் In English

14

The basic idea of the greedy approach is to calculate the ratio value/weight for each item and sort the item based on this ratio. Then take the item with the highest ratio and add it to the knapsack until we can't add the following item as a whole, and at the end, add the following item as much as possible.

28-Sept-2022

https://byjusexamprep.com › knaps...

Knapsack Problem Using Greedy Method - Detail, Algorithm, Example

About featured snippets Feedback

Algorithm

15

- **Step 1** : Get the two inputs, the positive value of n and the non-positive value of k which denotes the k-th binomial coefficient in the Binomial Expansion.
- **Step 2** : Allocate the array of size k + 1 with the value of 1 at 0-th index and rest with value 0.
- **Step 3** : Next, generating the sequence of pascal's triangle, with the first row containing single element valued 1 which was already created in step 2.
- **Step 4** : Further next consecutive rows of pascal's triangle are computed from the previous row by adding the two consecutive elements, but step 4 is to be carried out upto k-times, for enclosing n-value times.
- **Step 5** : Stop.

16

தமிழ் இல்

In English

The 0/1 knapsack problem means that **the items are either completely or no items are filled in a knapsack**. For example, we have two items having weights 2kg and 3kg, respectively. If we pick the 2kg item then we cannot pick 1kg item from the 2kg item (item is not divisible); we have to pick the 2kg item completely.

https://www.javatpoint.com › 0-1-k...

DAA | 0/1 Knapsack Problem - Javatpoint

17

தமிழ் இல்

In English

The two dynamic programming properties which can tell whether it can solve the given problem or not are: Optimal substructure: an optimal solution to a problem contains optimal solutions to subproblems. Overlapping subproblems: a recursive solution contains a small number of distinct subproblems repeated many times.

01-Dec-2022

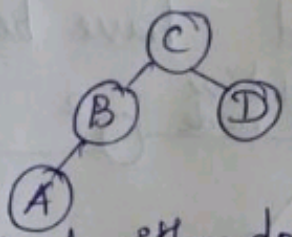
https://er.yuvayana.org › what-is-dy...

What is Dynamic Programming: Properties & examples - Yuvayana

18. OBST:

An Optimal Binary Search Tree is the one for which the average number of comparisons in a search is the smallest possible, if the probability of searching each elements is given.

Ex: For keys with A, B, C, D with probabilities 0.1, 0.2, 0.4 and 0.3.



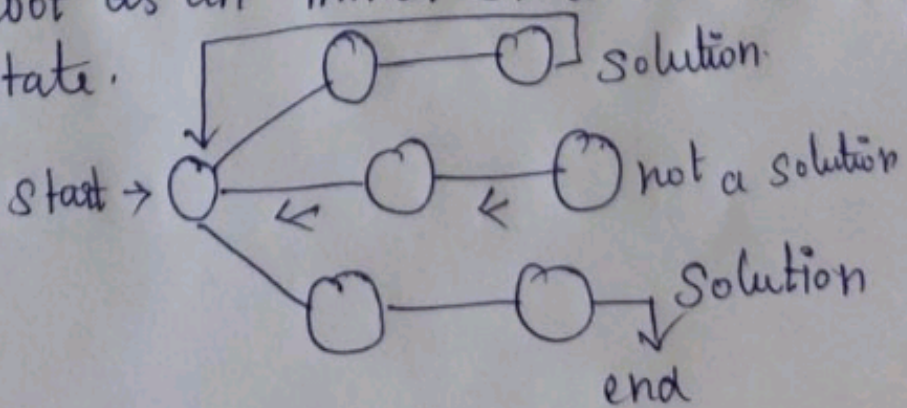
⇒ OBST.

19. What does Floyd's algorithm do?

Floyd's algorithm is an algorithm for finding the shortest path between all the pair of vertices in a weighted graph. This algorithm works for both the directed and undirected weighted graphs. But, it does not work for the graph with negative cycles.

20 What is a state space tree?

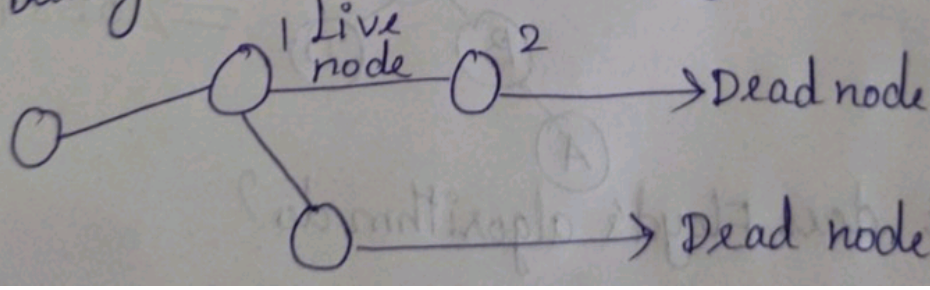
A space state tree is a tree representing all the possible states (solution or nonsolution) of the problem from the root as an initial state to the leaf as a terminal state.



21

Live node	Dead node.
* Live node represent that node which is been currently expanding	* Dead node represent a node which cannot be expanded further.
* Live nodes are able to acquire new links	* Dead nodes are static
* In this, all children are not yet been generated	* In this, all of its children have been generated.

* Ex:



22. Define Hamiltonian problem

* The Hamiltonian problem is about finding a path or cycle in a graph that visits every vertex exactly once.
* It asks whether such a path or cycle exists in a given graph.

23. Differentiate Backtracking and Branch and Bound

Backtracking	Branch and Bound
It is used to find all possible solution available to the problem	It is used to solve optimization problem
It traverse tree by DFS (Depth First Search)	It may traverse the tree in any manner DFS or BFS
It search the state space tree until it found a solution	It completely searches the state space tree to get optimal solution
It involves feasibility function	It involves bounding function

24. What is non-deterministic polynomial time.

It is ~~one~~ one of the best-known complexity classes in theoretical computer science. A decision problem is said to be in NP if it is solvable in polynomial time by a Turing non-deterministic Turing machine

▶ Algorithm 3.2

Binomial Coefficient Using Dynamic Programming

Problem: Compute the binomial coefficient.

Inputs: nonnegative integers n and k, where $k \leq n$.

Outputs: $bin2$, the binomial coefficient $\binom{n}{k}$.

```
int bin2 (int n, int k)
{
  index i, j;
  int B[0..n][0..k];

  for (i = 0; i <= n; i++)
     for (j = 0; j <= minimum(i,k); j++)
        if (j == 0 || j == i)
           B[i][j] = 1;
        else
           B[i][j] = B[i-1][j-1] + B[i-1][j];
  return B[n][k];
}
```

- **ALGORITHM** *Warshall(A*[1..*n*, 1..*n*]*)*

  //ImplementsWarshall’ s algorithm for computing the transitive closure

  //Input: The adjacency matrix *A* of a digraph with *n* vertices

  //Output: The transitive closure of the digraph

  $R^{(0)} \leftarrow A$

  for $k \leftarrow 1$ **to** n **do**

  for $i \leftarrow 1$ **to** n **do**

  for $j \leftarrow 1$ **to** n **do**

  $R^{(k)}[i, j] \leftarrow R^{(k-1)}[i, j]$ **or** $(R^{(k-1)}[i, k]$ **and** $R^{(k-1)}[k, j])$

  return $R^{(n)}$

- The time efficiency is only $\Theta(n^3)$.