1. Describe in-order, pre-order, and post-order traversals of a binary tree.

Answer: In an in-order traversal the left subtree is fully processed, then the root of the subtree is processed, then the right subtree is fully processed. In pre-order it is root, the left, then right. In post-order it is left, then right, then root.

2. What tree would result from entering the following numbers (in the order given) in an initially empty binary search tree: 54, 16, 13, 72, 43, 76, 86, 9, 24

A picture containing object

Description automatically generated

3. What would the tree from step 2 look like if you then deleted 16 from the tree?

A picture containing object

Description automatically generated

4. In a Red-Black tree, if the number of keys on the shortest simple path from root to leaf is n, what is the maximum number of keys that could appear on any other simple path from root to leaf?

Answer: 2n

5. What are the left and right rotations used for in a Red-Black tree

Answer: The left and right rotations are used to rebalance the height of the tree furing insertions and deletions.

6. In the dynamic programming algorithm for chain matrix multiplication, what data is memoized and what data structure(s) is used to store the memoization?

Answer: In the chain matrix multiplication algorithm the minimum number of arithmetic operations needed to multiply matrices the sequence of matrices from Mi to Mj was saved together with an index k denoting where the parentheses should go. The data was saved in a matrix denoting I and j.

7. What property must a problem have to be a candidate for a dynamic programming approach?

Answer: to use dynamic programming a problem must be able to be broken up into smaller subproblems which can be solved and the results of those subproblems used for solving bigger subproblems.

8. What does it mean if we say an algorithm is a greedy algorithm?

Answer: an algorithm is a greedy algorithm if there is some easy choice that can be taken at each step that will result in finding a solution.

9. Do greedy algorithms return optimal solutions?

Answer: It depends on the problem.

10. Suppose we have a picture that has the following colors and frequencies of pixels of each color (in thousands): Red: 47 Blue: 63 Green: 33 Yellow: 12 Orange: 17 Black: 5 While: 2

1. looking at the colors, does it look like it may be a good candidate for Huffman encoding?

Since there is a wide difference between the high and low frequencies, this is likely a good candidate.

1. What would the Huffman encoding be? (show your work)

A close up of a piece of paper

Description automatically generated

1. Does it save space over an encoding of 3 bits per color?

Yes. At 3 bits per color it would be 537000 bits. Using coding would be 420000 bits.

11. If a node in a B-tree has n keys in it and is not a leaf, how many children does it have?

Answer: A B-tree with n keys that is not a leaf has n+1 children.

12. In a B-tree, what does the branching factor, t, used to determine?

Answer: In a B-tree, the branching factor, t, determines the minimum and maximum number of keys. All nodes except the root can have a minimum of t-1 keys in it. All nodes including the root can have a maximum of 2t-1 keys.

13. The CS department averages between 100 and 150 graduate students at any given time. If we wanted to store information about students in a hash table that had hash lists with possibly some hash collisions (but maximum an average of < 2 per bin), give a hash function you believe would be good, indicate how many hash bins it would use, and state why you believe it would be a good hash function.

Answer: choose 100bins, hash function is R-Number mod 2. Assuming that the last 2 digits of R-Numbers are evenly distributed across students this will give an even distribution with less than 2 items per bin on average.