

1. For each traversal below, draw the corresponding binary search tree. Whenever any node has only one child, clearly show whether it is a left child or a right child.

- Preorder: L J E A C G N R P Y T

- Postorder: D F K H B Q W U S Z M

- Level-order: C V E X J H T P M R L S

2. For each traversal below, draw the corresponding arithmetic expression tree, and also write an equivalent infix expression using the *fewest* parentheses. Assume all operators are binary, and infix expressions are evaluated using C++ precedence and associativity rules.

- Prefix: $/ + - C * E G A \% * D + F H B$

- Postfix: $Q J L + N / \% R K M * P + - *$

- Level-order: $* - S Z + * T Y / \% U + V X W$

3. For each pair of traversals below, draw the corresponding general (non-binary) tree.

- Preorder: F K Q P J D A M C H L G E B N
Postorder: Q K J A D C H M G L E P N B F

- Preorder: R Y B F T C W Z D G A E S X V
Level-order: R Y B D V F G T C Z A E X W S

- Postorder: N K T Q X H W P J Z L R U M S
Level-order: S M H U N Q X P J Z R K T W L

4. Given the letters and frequencies below, draw the Huffman tree and write the Huffman code for each letter. At each step of the Huffman algorithm, always connect the smallest frequency node as the left child and the second smallest frequency node as the right child.

Letter	Freq	Code
A	8	
E	16	
H	5	
I	12	
L	6	
M	2	
N	10	
O	4	
R	13	
S	9	
T	14	
W	1	