Preorder traversal:

Visit the root first, then all subtrees (from left to right) in preorder.

Postorder traversal:

Visit first all subtrees (from left to right) in postorder, then visit roots.

Inorder traversal (binary trees only):

Visit the left subtree, then the root, then the right subtree.

Polish notation corresponds a preorder traversal of the binary tree representing an algebraic expression. Standard notation (with brackets) corresponds to inorder traversal.

Depth-first Search

INPUT: **A connected graph** $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$.
OUTPUT: **A (depth-first) spanning tree** T **of** G.

(*Step 1*) **Initialize** $T = (V_T, E_T)$: $V_T = \{v_1\}$ **and** $E_T = \emptyset$. **Set** $current := v_1$.

(*Step 2*) **Select the neighbour of** *current* **with the lowest index which is not yet visited. If no such neighbour exist, continue to** *Step 3*. **If it does exist, say** v_i, **set** $current := v_i$ **and return to** *Step 2*.

(*Step 3*) **If** $current \neq v_1$, **then find the parent** v_j **of** *current*, **and set** $current = v_j$ **(backtracking). Go to** *Step 2*.

(*Step 4*) **If** $current = v_1$, **then stop.**

Breadth-first Search

INPUT: **A connected graph $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$.**
OUTPUT: **A (breadth-first) spanning tree T of G.**

(*Step 1)* **Initialize $T = (V_T, E_T)$: $V_T = \{v_1\}$ and $E_T = \emptyset$. Insert v_1 in a queue Q.**

(*Step 2)* **Set *current* to be the vertex at the front of the queue Q, and delete this vertex from Q. Add all neighbours of *current* that have not been visited to the rear of the queue, in order of increasing index. If no such neigbours exist, repeat *Step 2*.**

(*Step 3)* **If Q is empty, then stop.**

Vertex set: $\{a, b, c, d, e, f, g, h\}$.
Adjacency matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$