Problem 1. Fill in the blanks regarding the worst-case running times for the following algorithms and functions on a binary search tree with \( n \) nodes and height \( h \).

- The running time of insertion is \( \Theta(\text{function of } n \text{ and } h) \).
- The running time of function \texttt{max} in Problem 1 of \texttt{pqTREE1} is \( \Theta(\text{function of } n \text{ and } h) \).
- The running time of function \texttt{after} in Problem 2 of \texttt{pqTREE1} is \( \Theta(\text{function of } n \text{ and } h) \).
- The running time of function \texttt{fill_sorted} in Problem 3 of \texttt{pqTREE1} is \( \Theta(\text{function of } n \text{ and } h) \).

Problem 2. Determine the truth of the following statements about BSTs with \( n \) nodes and height \( h \).

- The worst-case running time of searching is \( \Theta(n + h) \). □ True □ False
- The running time of printing all elements is \( \Theta(n) \). □ True □ False
- The worst-case running time of insertion is \( \Theta(\log(n)) \) □ True □ False
- \( \lfloor \log_2(n) \rfloor \leq h \leq n \) □ True □ False
**Problem 3.** Complete the following template function that prints only the internal nodes of a BST.

```cpp
template <typename T>
void print_internal_nodes(Node<T>* root) {
    if (root == nullptr)
        _____;
    if (root->left _____ _____)
        cout << root->x << endl;
    else if (root->right _____ _____)
        cout << root->x << endl;
    print_internal_nodes(root->left);
    print_internal_nodes(root->right);
}
```

For trees with \( n \) nodes, the running time of \( \text{print\_internal\_nodes} \) is \( \Theta(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) \).