

1. Hàm `insert(int father, int data)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng và thêm phần tử đầu tiên.
- Trường hợp xấu nhất: $O(n)$ - Khi phải duyệt qua toàn bộ các anh em của một node cha để tìm vị trí chèn node mới (vì có thể có đến n node anh em).

2. Hàm `findNode(Node* node, int data)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi node cần tìm là node gốc.
- Trường hợp xấu nhất: $O(n)$ - Khi node cần tìm là node cuối cùng hoặc không có trong cây. Khi đó, phải duyệt qua toàn bộ các node trong cây.

3. Hàm `remove(int data)` và `removeHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi node cần xóa là node lá, không có con.
- Trường hợp xấu nhất: $O(n)$ - Khi node cần xóa là node gốc và có nhiều con, cần xóa toàn bộ cây.

4. Hàm `preorder()` và `preorderHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.
- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua toàn bộ các node trong cây để in ra theo thứ tự duyệt trước.

5. Hàm `postorder()` và `postorderHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.
- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua toàn bộ các node trong cây để in ra theo thứ tự duyệt sau.

6. Hàm `isBinaryTree()` và `isBinaryTreeHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.
- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua tất cả các node và kiểm tra số lượng con của từng node.

7. Hàm `isBinarySearchTree()` và `isBinarySearchTreeHelper(Node* node, int min, int max)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.
- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua toàn bộ các node trong cây để kiểm tra điều kiện cây nhị phân tìm kiếm.

8. Hàm `isMaxHeapTree()` và `isMaxHeapHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.
- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua toàn bộ các node trong cây để kiểm tra điều kiện của cây max-heap.

9. Hàm `height()` và `heightHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.
- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua toàn bộ các node trong cây để tính chiều cao của cây.

10. Hàm `findMax()` và `findMaxHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.
- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua toàn bộ các node trong cây để tìm giá trị lớn nhất.

11. Hàm `numOfLeaves()` và `numOfLeavesHelper(Node* node)`

Độ phức tạp:

- Trường hợp tốt nhất: $O(1)$ - Khi cây rỗng.

- Trường hợp xấu nhất: $O(n)$ - Phải duyệt qua toàn bộ các node trong cây để đếm số lượng node lá.