## 題目敘述
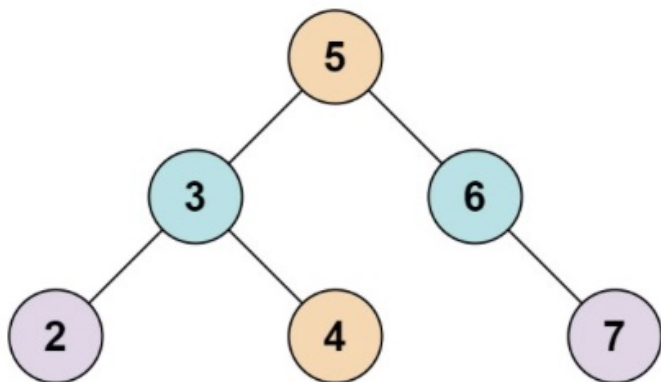
Given the `root` of a binary search tree and an integer `k`, return `true` if there exist two elements in the BST such that their sum is equal to `k`, or `false` otherwise.
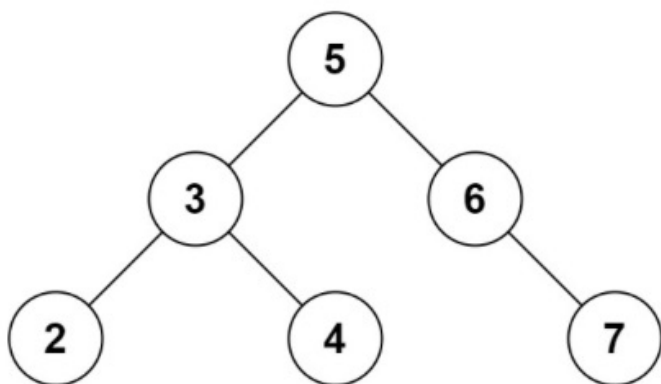
**Example 1:**



**Input:** root = [5,3,6,2,4,null,7], k = 9
**Output:** true

**Example 2:**



**Input:** root = [5,3,6,2,4,null,7], k = 28
**Output:** false

## 參考答案

```cpp
struct Node {
    int val;
    Node *left;
    Node *right;
    Node() : val(0), left(nullptr), right(nullptr) {}
    Node(int x) : val(x), left(nullptr), right(nullptr) {}
    Node(int x, Node *left, Node *right) : val(x), left(left), right(right) {}
};

class Solution {
private:
    vector<int> v;
```

```cpp
    void _solve (Node *st) {
        if (!st) return;
        _solve(st→left);
        _solve(st→right);
        v.emplace_back(st→val);
    }

public:
    bool solve(Node* root, int k) {
        map<int, int> mp;
        _solve(root);
        for (int &i: v) ++mp[i];
        for (int &i: v) {
            if (k == 2 * i) {
                if (mp.find(k - i) ≠ mp.end() and mp[k - i] > 1) return true;
            }else if (mp.find(k - i) ≠ mp.end()) return true;
        }
        return false;
    }
};
```