

**Level: Easy****Find if a given number N exists in a stack S.**

Note: This question too easy for interviews. It is only to demonstrate traversing a stack.

**Questions to Clarify:**

Q. What output should I return?

A. Return true if N is found, false otherwise.

Q. So I return true as soon as I find one instance of the number?

A. Yes

**Solution:**

We traverse through the stack using a temporary stack. As soon as we find a node that is equal to N, we return. Before returning, we restore the original stack.

**Pseudocode:**

(Note: Never write pseudocode in an actual interview. Unless you're writing a few lines quickly to plan out your solution. Your actual solution should be in a real language and use good syntax.)

```
find(N, S)
    Stack temp = empty
    boolean found = false
    while s is not empty
        top = s.peak()
        if top == N
            found = true
            break
        temp.push(s.pop())

    // restore S
    while temp is not empty
        s.push(temp.pop())

    return found
```

**Test Cases:**

Edge Cases: empty stack, null stack

Base Cases: one item in stack, 2 items in stack (N found/not-found)

Regular Cases: many items in stack (N found/not-found)

Time Complexity:  $O(n)$

Space Complexity:  $O(1)$

```
public static boolean find(int target, Stack<Integer> s) {
    if (s == null)
        return false;

    Stack<Integer> temp = new Stack<>();
    boolean found = false;

    while (!s.isEmpty()) {
        if (s.peek() == target) {
            found = true;
            break;
        }

        temp.push(s.pop());
    }

    while (!temp.isEmpty()) {
        s.push(temp.pop());
    }

    return found;
}
```