2: Searching: Sample

Sample Solution
31st July 2025
Collaborators: None

1: "Search for an element in a list using a while loop"

BLUEPRINT

Requires:

List l : 'a list

Target t : 'a

Ensures:

If $\exists i \in [0, length(l) - 1]$ such that $l[i] = t$ then return True
Else $\forall i \in [0, length(l) - 1]$, if $l[i] <> t$ then return False

Time Complexity: O(n)

Space Complexity: O(1)

Input: Input: l = [1; 3; 5; 6], t = 5
Output: Output: True
Input: Input: l = ["AX67", "HD09", "FGSJ", "78AD"], t = "AB01"
Output: Output: False
Input: Input: l [True, False, True, False, False], t = 1
Output: Output: False

STEPS

Step 1: In order to check for indices, we need a counter. Let's make this i

Step 2: We want to ensure that the counter is never >= the length. This is erronous. This can be one condition of the while loop.

Step 3: We also want that if the element exists at i, we do not loop anymore. This gives us our 2 conditions for the loop to continue.

Step 4: If we enter the loop, the only thing that is left to do is increment i so that we can check for the next index. If the next index is valid and the element is not at the next index, the loop breaks and so on.

Step 5: Our truth value is hidden in at what point the loop terminates.

Case 1: The loop terminates because of the length condition means the element was not found

Case 2: The loop terminates because l[i] = t means the element was found

OCAML CODE ICS Verified

```
let search(l : 'a list)(t : 'a) : bool =
    let i = ref 0 in
    let len = ref( List.length l )in
        while i < len && (List.nth l !i) <> t do
            i := !i + 1;
        done;
```

```
          !i <> !len
      ;;
```

PROOF

INVARIANT

Invariant Statement:

i (counter) = ref 0

Inv(i) (Invariant): !i < List.length l && l[i] <> t

Maintenance:

Inv(i) = !i < List.length l && l[i] <> t

At (i+1): Case 1: !i == List.length l: Ensures $\forall i \in [0, List.length l - 1] \neq t$, return !i < List.length l = false

Case 2: (!i < List.length l && List.nth l i = t) = true: Ensures that $\exists i \in [0, List.length l - 1]$ such that List.nth l i = t, return !i < List.length l = True

Case 3: !i < List.length l && List.nth l i <> t: i := !i + 1, Inv(i+1) holds

Therefore, Inv(i) => Inv(i+1)

Termination:

Case 1: Loop is terminated becasue !i == List.length l: Ensures $\forall i \in [0, List.length l - 1] \neq t$, return !i < List.length l = false

Case 2: Loop is terminated !i < List.length l && List.nth l i = t) = true: Ensures that $\exists i \in [0, List.length l - 1]$>/tex> such that List.nth l i = t, return !i < List.length l = True

Either of these cases must happen. Consider Case 2 never occurs. Then, Case 1 is eventually reached as i := !i + 1

Consider Case 1 never occurs. Then Case 2 must have occured for the loop to have been exited before !i == List.length l.

Therefore, the loop termintes.