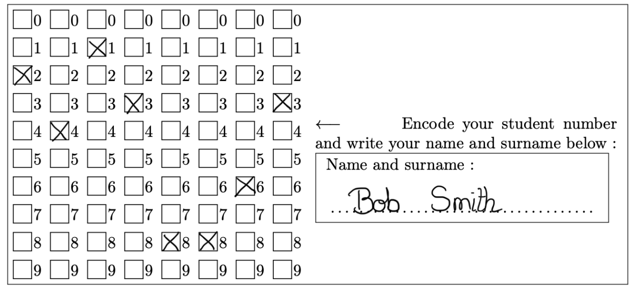
Data Structures TD4: Mock Exam

Amandine Decker & Marie Cousin

December 2023

1 Questions

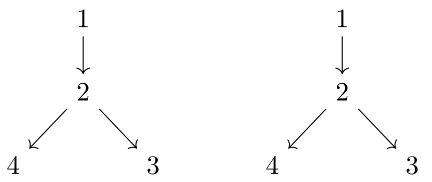
Fill the questionnaire by ticking or colouring the box corresponding to the right answer. Write your student number in the first box, one number per column (see right: Bob Smith, number 24138863). Use a pencil to be able to correct mistakes!



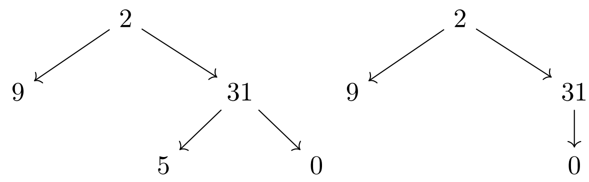
2 Application of an Algorithm

```
Require: T1, T2    -- We need two trees as input
if isEmpty(T1) and isEmpty(T2) then
    return True
else if isEmpty(T1) or isEmpty(T2) then
    return False
end if
S1 ← ⊢ T1 ⊢
S2 ← ⊢ T2 ⊢
while !isEmpty(S1) and !isEmpty(S2) do
    tree1 ← get(S1)
    tree2 ← get(S2)
    remove(S1)
    remove(S2)
    if tree1['root'] != tree2['root'] then
        return False
    end if
    if !isEmpty(tree1['left child']) and !isEmpty(tree2['left child']) then
        add(tree1['left child'], S1)
        add(tree2['left child'], S2)
    else if !isEmpty(tree1['left child']) and isEmpty(tree2['left child']) or
isEmpty(tree1['left child']) and !isEmpty(tree2['left child']) then
        return False
    end if
    if !isEmpty(tree1['right child']) and !isEmpty(tree2['right child']) then
        add(tree1['right child'], S1)
        add(tree2['right child'], S2)
    else if !isEmpty(tree1['right child']) and isEmpty(tree2['right child']) or
isEmpty(tree1['right child']) and !isEmpty(tree2['right child']) then
        return False
    end if
end while
return True
```

1. Apply the algorithm above to the pairs of trees (a) and (b):



(a)



(b)

Applying the algorithm means writing down the different states of the variables.

2. Explain what the above algorithm does (*i.e.*, the idea of the algorithm) with your own words.

3 Writing an Algorithm

1. **All**: Write an algorithm that takes a dictionary such that its keys are strings and its values are booleans as input and outputs $True$ if all of its values (in the key-value pairs) are $True$, and $False$ otherwise (if the dictionary is empty, return $True$). For example, this algorithm should output $False$ with the input $\{(`bird', True), (`frog', False),$ $(`horse', True), (`lion', True)\}$ and $True$ with the input $\{(`bird', True), (`frog', True), (`horse', True), (`lion', True)\}$.

2. **Sort**: Write an algorithm that takes a *stack* containing positive integers as input and outputs a *list* containing all the elements of the stack in ascending order. For example this algorithm should output $[3, 4, 5, 6, 9]$ with the input $\vdash 5, 6, 9, 3, 4\ \vdash$, and $[]$ with the input $\vdash\ \vdash$.

3. **Sorted concatenation**: Write an algorithm that takes two sorted lists of integers (sorted from smallest to largest) as input and outputs a sorted list that contain all the elements from both lists. For example, this algorithm should output $[1, 2, 3, 3, 4, 5]$ with the input $[1, 3, 5]$, $[2, 3, 4]$.