

128



Coláiste na hOllscoile
Corcaigh
University College Cork

25

[a] ✓

15

for n lbst

rbst

Uimhir Scrúdaithe
Examination Number

9 1 7 1 6 3

Module Code CS4621

Paper No. _____

Mír
Section _____

Do na Scrúdaitheoirí amháin
For Examiner's use only

1	25
2	34
3	33
4	✓
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
Iomlán Total	92% 64

Calculator, Please state:	No. of Books submitted
Name	7
Model	

Note: If there are different sections on this paper,
a separate Answer Book MUST be used for each section.

Q1

25

a

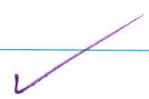
ancestors :: Ord a => a -> BST a -> [a] ✓

15

ancestors n (Node (Lbst) v (Rbst)) | n < v = v : ancestors n Lbst

n > v = v : ancestors n Rbst

n == v = v : []
[v]



Q1

B $cca :: Ord a \Rightarrow a \rightarrow a \rightarrow BST a \rightarrow Bst a$ ✓

$cca n1 n2 (Node lbst v rbst)$

$|(Occurs n1 lbst) \&\& (Occurs n2 lbst) = cca n1 n2 lbst$

$|(Occurs n1 rbst) \&\& (Occurs n2 rbst) = cca n1 n2 rbst$

$| \text{Otherwise} = \text{Node lbst v rbst}$

correct, but very inefficient

10

--Occurs n bst does 'n' occur in the bst 'bst'

$Occurs :: Ord a \Rightarrow a \rightarrow BST a \rightarrow Bool$

$Occurs n \text{ Empty BST} = False$

$Occurs n (Node (lbst) v (rbst)) \mid n < v = Occurs n lbst$

$\mid n > v = Occurs n rbst$

$\mid n == v = True$

If one really wanted to use ancestors they could

call it on both, find the furthest to the end item

in the 2 lists, that's our common ancestor and

now locate this subtree in our BST ✓

Q2

A

34

~~24~~

type Stack a = [a]

emptyStack = []

isEmptyStack [] = True

isEmptyStack _ = False

Push x s = x : s

Pop (_ : ros) = ros

~~15~~

14

top (s : _) = s

length (_ : ros) = 1 + length ros

length [] = 0

size

name 'length' would clash with builtin name

Q2

B type Stack a = ([a], Int)

emptyStack = ([], 0) -

isEmptyStack ([], 0) = True

isEmptyStack _ = False

20

push x (s, c) = ((x:s), c+1)

pop ((_:ros), c) = (ros, c-1)

top (s:_, _) = s

length (_, c) = c

size

Q3

33

a get Def t ((dt, dv) : rEnv) | t == dt = dv
| otherwise = getDef t rEnv

b

* If definition not in Env we get error *

Q 3

Continues \rightarrow

b

eval' IF ts $((S_{quotation} q) : (S_{quotation} -) ; (S_{boolean}$

eval' IF ts $((S_{quotation} -) ; (S_{quotation} q) : (S_{boolean}$

Q3 B

True): pos } env = eval' (q ++ ts) (ros) (env)
False): pos } env = eval' (q ++ ts) (ros) (env)

7

Q3

C The most straightforward way of implementing it is as follows:

In eval' after the definition of IF have the following

Modify eval'

IF t == "IF" then

else

IF t == "call" then

eval' call ts stack env

else

case t of

Define function eval' call

-- eval' call tokens stack env, Execute Quotation at the top of stack

20

eval' call :: [Token] -> Stack -> Environment -> String

eval' call ts ((quotation w); ros) env =
eval' (q++ ts) (ros) (env)