floor ("p")

Node x = floor (root, "p")
return root.right.left.left.left

Nodes examined: M, T, R, Q, O

M floor (root, "p")
Node t = floor(root.right, "p")
return root.right.left.left.left

I floor (root.right, "p")
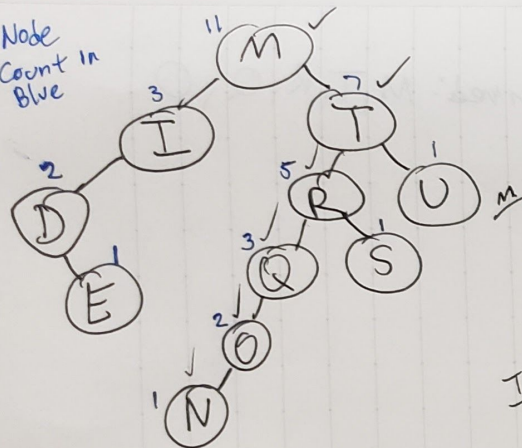return floor (root.right.left, "p")

R floor(root.right.left, "p")
return floor(root.right.left.left, "p")

Q floor (root.right.left.left, "p")
return floor (root.right.left.left.left, "p")

O floor(root.right.left.left.left, "p")
Node t = floor (root.right.left.left.left.right, "p")
return root.right.left.left.left

null floor (root.right.left.left.left.right, "p")
return null

Node Count in Blue

[Tree diagram with nodes: M(11) at top, I(3) and T(7) below. I has D(2), T has R(5) and U(1). D has E(1). R has Q(3) and S(1). Q has O(2). O has N(1).]

Select(5)
return select(root, 5)   Nodes examined: M, T, R, Q, O, N

root.right.left.left.left

M
Select(root, 5)
  left Size = 3
  (3 < 5)
    return select(root.right, 5-3-1)

I
Select(root.right, 1)
  left Size = 5
  (5 > 1)
    return select(root.right.left, 1)

R
Select(root.right.left, 1)
  left Size = 3
  (3 > 1)
    return select(root.right.left.left, 1)
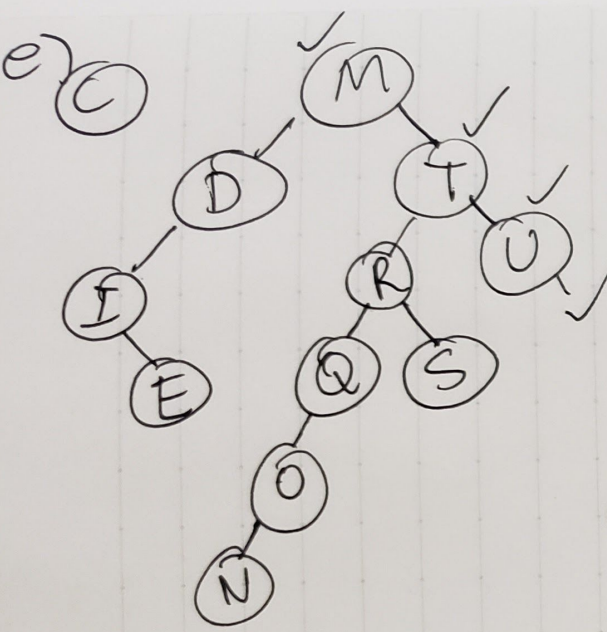
Q
Select(root.right.left.left, 1)
  left Size = 2
  (2 > 1)
    return select(root.right.left.left.left, 1)

O
Select(root.right.left.left.left, 1)
  left Size = 1
    return root.right.left.left.left

e) b

Ceiling ("U")
Node x = ceiling(root, "U")
```
┌─────────────────────────────┐
│ Error, argument to ceiling() │
│ too large                    │
└─────────────────────────────┘
```
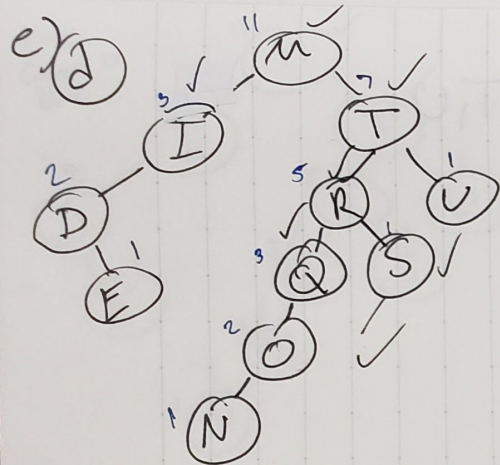
Ceiling(root, "U")
  return ceiling(root.right, "U")

Ceiling(root.right, "U")
  return ceiling(root.right.right, "U")

ceiling(root.right.right, "U")
  return ceiling(root.right.right.right, "U")

ceiling(root.right.right.right, "U")
  return null

Nodes examined: M, T, U

Node count in blue

rank("S")
    return rank("S", root) → 8

rank("S", root)
    return 1 + size(root.left) + rank("S", root.right) → 8
                   3   4

rank("S", root.right)
    return rank("S", root.right.left)

rank("S", root.right.left)
    return 1 + size(root.right.left.left) + rank("S", root.right.left.right) → 4
                     3   0

rank("S", root.right.left.right)
    return size(root.right.left.right.left)

size(root.right.left.right.left)
    return 0

size(root.right.left.left)
    return 3

size(root.left)
    return 3

Nodes Examined: M, I, T, R, Q, S,