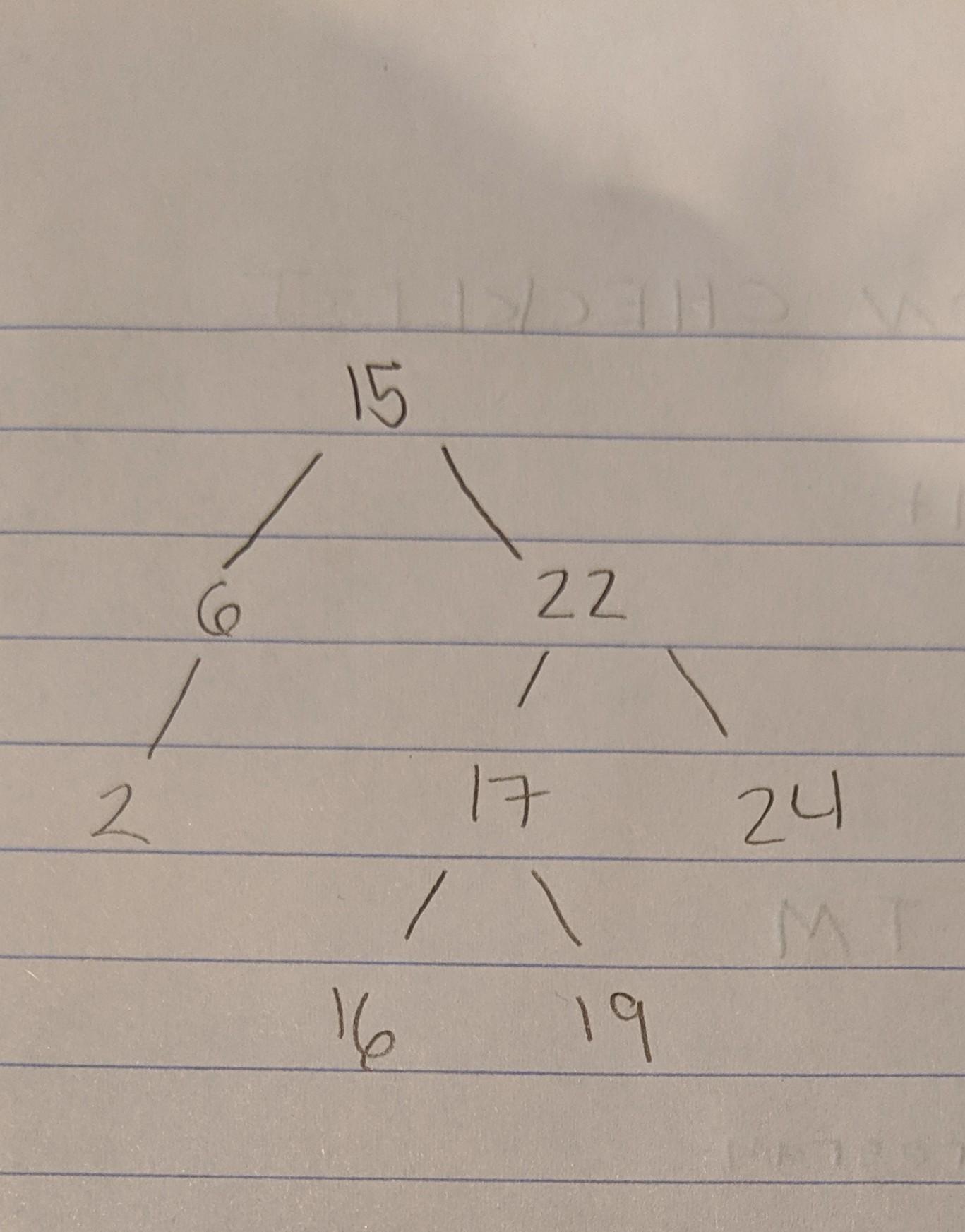
**0.**

****

**1.**

(define (contains? bst num)

(cond ((null? bst) #f)

((> num (datum bst)) (contains? (right bst) num))

((< num (datum bst)) (contains? (left bst) num))

(else #t)))

**2.**

(define (in-order bst)

(if (empty-tree? bst)

'()

(append (in-order (left bst))

(list (datum bst))

(in-order (right bst)))))

**3.**

(define (count-nodes bst)

(if (empty-tree? bst)

0

(+ 1 (count-nodes (left bst)) (count-nodes (right bst)))))

**4.**

(define (square-tree bst)

(if (empty-tree? bst)

tet

(make-bintree (square (datum bst))

(square-tree (left bst))

(square-tree (right bst)))))