binary trees

```
; a bt is either
;  - empty
;  - (make-node number bt bt)
(define-struct node (num l r))

; find : bt number -> boolean
; to determine if n is in bt
(define (find bt n)
  (cond
    [(empty? bt) false]
    [else (or (= n (bst-num bt))
              (find (bst-l bt) n)
              (find (bst-r bt) n))]))
```

```
(find                          =>
 (make-bt
   5
   (make-bt
     3
     empty
     empty)
   (make-bt
     8
     (make-bt
       7
       empty
       empty)
     empty))
 7)
```

```
(find
 (make-bt
   5
   (make-bt
     3
     empty
     empty)
   (make-bt
     8
     (make-bt
       7
       empty
       empty)
     empty))
 7)
```

```
=> (or false
       (find
        (make-bt
          3
          empty
          empty)
        7)
       (find
        (make-bt
          8
          (make-bt
            7
            empty
            empty)
          empty)
        7))
```

```
(or false                    =>
    (find
     (make-bt
       3
       empty
       empty)
      7)
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
      7))
```

```
(or false
    (find
     (make-bt
       3
       empty
       empty)
      7)
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
      7))
```

```
=> (or false
       (or false
           (find empty 7)
           (find empty 7))
       (find
        (make-bt
          8
          (make-bt
            7
            empty
            empty)
          empty)
         7))
```

```
(or false                    =>
    (or false
        (find empty 7)
        (find empty 7))
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
      7))
```

```
(or false                    => (or false
    (or false                        (or false
        (find empty 7)                   false
        (find empty 7))                  (find empty 7))
    (find                            (find
     (make-bt                         (make-bt
       8                                8
       (make-bt                         (make-bt
         7                                7
         empty                            empty
         empty)                           empty)
       empty)                           empty)
     7))                              7))
```

```
(or false                    =>
    (or false
        false
        (find empty 7))
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
      7))
```

```
(or false                          => (or false
    (or false                              (or false
        false                                  false
        (find empty 7))                        false)
    (find                                  (find
     (make-bt                               (make-bt
       8                                      8
       (make-bt                               (make-bt
         7                                      7
         empty                                  empty
         empty)                                 empty)
       empty)                                 empty)
     7))                                    7))
```

```
(or false                    =>
    (or false
        false
        false)
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
      7))
```

```
(or false
    (or false
        false
        false)
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
     7))
```

```
=> (or false
       false
       (find
        (make-bt
          8
          (make-bt
            7
            empty
            empty)
          empty)
        7))
```

```
(or false                    =>
    false
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
      7))
```

```
(or false
    false
    (find
     (make-bt
       8
       (make-bt
         7
         empty
         empty)
       empty)
     7))
```

```
=> (or false
       false
       (or false
           (find
            (make-bt
              7
              empty
              empty)
            7)
           (find
            empty
            7)))
```

```
(or false                    =>
    false
    (or false
        (find
         (make-bt
           7
           empty
           empty)
         7)
        (find
         empty
         7)))
```

```
(or false
    false
    (or false
        (find
         (make-bt
           7
           empty
           empty)
         7)
        (find
         empty
         7)))
```

```
=> (or false
       false
       (or false
           (or true
               (find
                empty
                7)
               (find
                empty
                7))
           (find
            empty
            7)))
```

```
(or false                    =>
    false
    (or false
        (or true
            (find
             empty
             7)
            (find
             empty
             7))
        (find
         empty
         7)))
```

```
(or false
    false
    (or false
        (or true
            (find
             empty
             7)
            (find
             empty
             7))
        (find
         empty
         7)))
```

```
=> (or false
       false
       (or false
           true
           (find
            empty
            7)))
```

```
(or false                    =>
    false
    (or false
        true
        (find
         empty
         7)))
```

```
(or false                    => (or false false true)
    false
    (or false
        true
        (find
         empty
         7)))
```

```
(or false false true)   =>
```

```
(or false false true)   => true
```

binary search trees

```
; a bst is either
;  - empty
;  - (make-node number[n] bst[l] bst[r])
; INV: n is larger than the numbers in l and smaller than those in r
(define-struct node (num l r))

; find : bst number -> boolean
(define (find bt n)
  (cond
    [(empty? bt) false]
    [else
     (cond
       [(= n (bst-num bt)) true]
       [(> n (bst-num bt)) (find (bst-l bt) n)]
       [(< n (bst-num bt)) (find (bst-r bt) n)])]))
```

```
(find          =>
 (make-bt
   5
   (make-bt
     3
     empty
     empty)
   (make-bt
     8
     (make-bt
       7
       empty
       empty)
     empty))
 7)
```

```
(find                    => (find
 (make-bt                    (make-bt
   5                           8
   (make-bt                    (make-bt
     3                           7
     empty                       empty
     empty)                      empty)
   (make-bt                    empty)
     8                       7)
     (make-bt
       7
       empty
       empty)
     empty))
 7)
```

```
(find          =>
 (make-bt
   8
   (make-bt
     7
     empty
     empty)
   empty)
 7)
```

```
(find                  => (find
 (make-bt                   (make-bt
   8                          7
   (make-bt                   empty
     7                        empty)
     empty                 7)
     empty)
   empty)
 7)
```

```
(find          =>
 (make-bt
   7
   empty
   empty)
 7)
```

```
(find          => true
 (make-bt
   7
   empty
   empty)
 7)
```