Questions 1-6

Question 1

1/1 point (graded)

Using the `check-expects` we designed in the last section, let's fill in the cells of the table

```
   p         bt | false | (make-node Nat Str BT BT)
---------------|-------|---------------------------
empty           |  (1)  |            (4)
---------------|-------|---------------------------
(cons "L" Path) |  (2)  |            (5)
---------------|-------|---------------------------
(cons "R" Path) |  (3)  |            (6)
```

What should the result of case (1) be?

- `true`
- **`false`** ✔
- `(has-path? <left child> (rest path))`
- `(has-path? <right child> (rest path))`

✔

Explanation

Recall the function should produce `false` if the path leads to a false BT.

Submit

Answers are displayed within the problem

Question 2

1/1 point (graded)

What should the result of case (2) be?

- `true`
- **`false`** ✔
- `(has-path? <left child> (rest path))`
- `(has-path? <right child> (rest path))`

✔

Explanation

Recall the function should produce `false` if the path leads to a false BT. This is the

Submit

Answers are displayed within the problem

Question 3

1/1 point (graded)

What should the result of case (3) be?

- `true`
- **`false`** ✔
- `(has-path? <left child> (rest path))`
- `(has-path? <right child> (rest path))`

✔

Explanation
Recall the function should produce `false` if the path leads to a false BT.

Submit

Answers are displayed within the problem

Question 4

1/1 point (graded)
What should the result of case (4) be?

- **`true`** ✔
- `false`
- `(has-path? <left child> (rest path))`
- `(has-path? <right child> (rest path))`

✔

Explanation
If the path is `empty` and the tree is not `false`, then the function produces `true` because the path leads to a node.

Submit

Answers are displayed within the problem

Question 5

1/1 point (graded)
What should the result of case (5) be?

- `true`
- `false`
- **`(has-path? <left child> (rest path))`** ✔
- `(has-path? <right child> (rest path))`

✔

Explanation
If the tree is not `false`, and the path starts with "L", then we need to recursively call the function on the left sub-tree.

Submit

Answers are displayed within the problem

Question 6

1/1 point (graded)

What should the result of case (6) be?

- `true`
- `false`
- `(has-path? <left child> (rest path))`
- `(has-path? <right child> (rest path))` ✔

Explanation
If the tree is not `false`, and the path starts with "R", then we need to recursively call the function on the right sub-tree.

Submit

Answers are displayed within the problem

Some Rights Reserved