Question 6

Bookmark this page

Question 6

1/1 point (graded)

After completing the design of `num-rooms`, you go to bed satisfied, feeling very proud of yourself. However you wake up the next morning with the nagging feeling that you could have designed `num-rooms` without adding an extra accumulator.

How could you have designed num rooms without an extra accumulator?

- produce `todo` at the end
- produce `visited` at the end
- produce `(lenth todo)` at the end
- produce `(length visited)` at the end ✔

Explanation

Once the todo list is empty, `(length visited)` will be the total number of rooms reached from the given room. This version of `num-rooms` will look like this:

```
(define (num-rooms r0)
  ;; todo is (listof Room); a worklist accumulator
  ;; visited is (listof String); context preserving acc, names of rooms already visited
  (local [(define (fn-for-room r todo visited)
            (if (member (room-name r) visited)
                (fn-for-lor todo visited)
                (fn-for-lor (append (room-exits r) todo)
                            (cons (room-name r) visited))))
          (define (fn-for-lor todo visited)
            (cond [(empty? todo) (length visited)]
                  [else
                   (fn-for-room (first todo)
                                (rest todo)
                                visited)]))]
    (fn-for-room r0 empty empty)))
```

Submit Show Answer

Answers are displayed within the problem