632474
Identify the bug in the following Java code (if any):

```
public boolean search(T item,ListADT<T> list){ // 1

   if (list == null) // 2
      return false; // 3
   else if (list.first()==item) // 4
      return true; // 5
   else // 6
      return this.search(item, list.rest()); // 7
} // 8
```

a. There is no base case

b. The problem is not self-similar

c. The problem does not get smaller

*d. There are no bugs

e. None of the above

f. "
g. "
h. "
i. "
j. "

General Feedback:

For a recursive solution to a problem, you need three things:
(1) a base case (where the problem can be solved without recursion)
(2) a self-similar problem (one that contains similar problem(s) to itself)
(3) a way of making the problem smaller so you get closer to the base case

Here (2) is satisfied -- lists contain smaller lists. (1) is satisfied by lines 2-5 of the method. And (3) is satisfied because line 7's recursive call takes the rest of the list as a parameter, rather than the whole list.

634918
Suppose QueueADT is implemented using a singly linked list. What is the lowest Big-Oh time complexity that can be achieved for an enqueue method?:

*a. O(1)

b. O(log$_2$ n)

c. O(n)

d. O(n^2)

e. none of the above

f. "
g. "
h. "
i. "

j. "

General Feedback:

If you keep a pointer to the last element of the queue, then the enqueue method just needs a fixed number of statements, which are the same independent of the size of the queue.

633554
What terminates a failed linear probe in a full hashtable?

a. The end of the array

b. A deleted node

c. A null entry

d. A node with a non-matching key

*e. Revisiting the original hash index

f. "
g. "
h. "
i. "
j. "

General Feedback:

A null entry will not appear in a full hashtable. Seeing the end of the array isn't correct, since we need to examine all elements, including those that appear before our original hash index. A node with a non-matching key is what started our probe in the first place. The purpose of leaving a deleted node in the table is so that probing may proceed past it. Revisiting the original hash index means we've looked at every entry and determined the item doesn't appear in the table.