UNIVERSITY OF CALIFORNIA, BERKELEY
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS10 Fall 2026 **TA: Nahee J.**



SOLUTIONS

Discussion 8: Tree Recursion + Algorithmic Complexity

Instructions:

- ~~If you're attending this section in-person, please log into iClicker!~~
- ~~If you missed this discussion, fill out this entire worksheet, and upload it to the Gradescope assignment titled "Discussion 8" by next Discussion.~~
- ~~Please open up snap.berkeley.edu/run on your computer.~~
- ~~For the worksheet, you can either explain the process in words, show a screenshot, or draw the block/process.~~

Group Activity / Question of the Day

- In groups of two or more, discuss your best study tips. What study techniques do you use when prepping for exams? Do you have a worst / least helpful technique you've come across? Do you usually study alone or in groups?

Required (Pages 2 - 3):

Section I - Tree Recursion

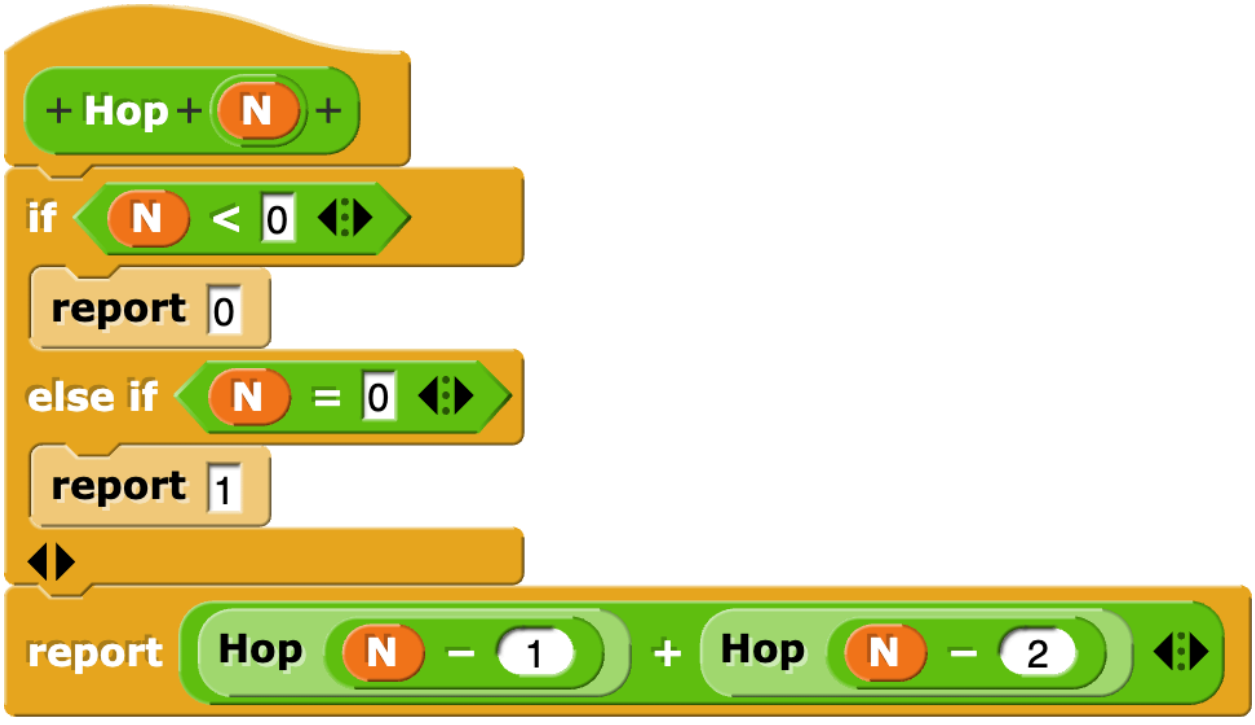
---

1. Hop Hop: We're climbing a staircase with a total of N steps. We can take either 1 or 2 steps at a time (a mix of both is possible) to reach the top. Write a procedure that takes in N and returns the number of ways that we can climb the stairs.

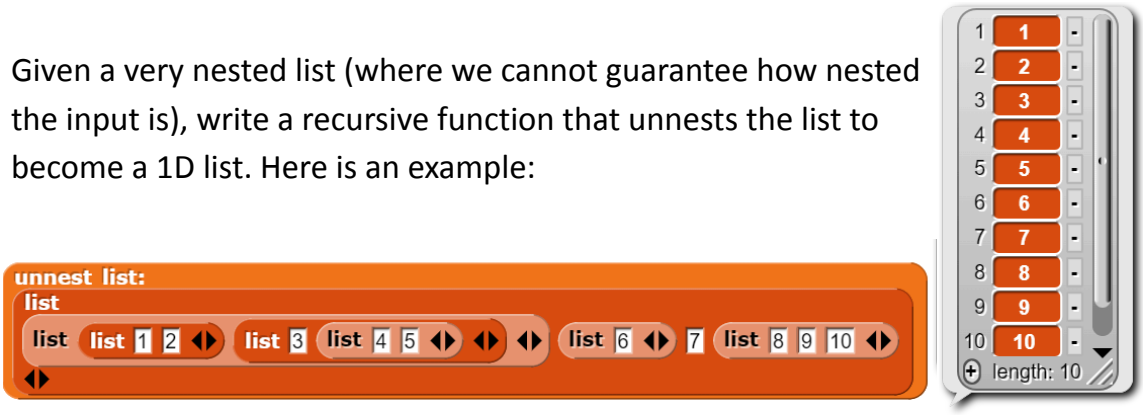
   For example, if N = 3, then you can climb the stairs in the following ways:

   a. Take 1 step, 1 step, 1 step
   b. Take 1 step, then 2 steps
   c. Take 2 steps, then 1 step

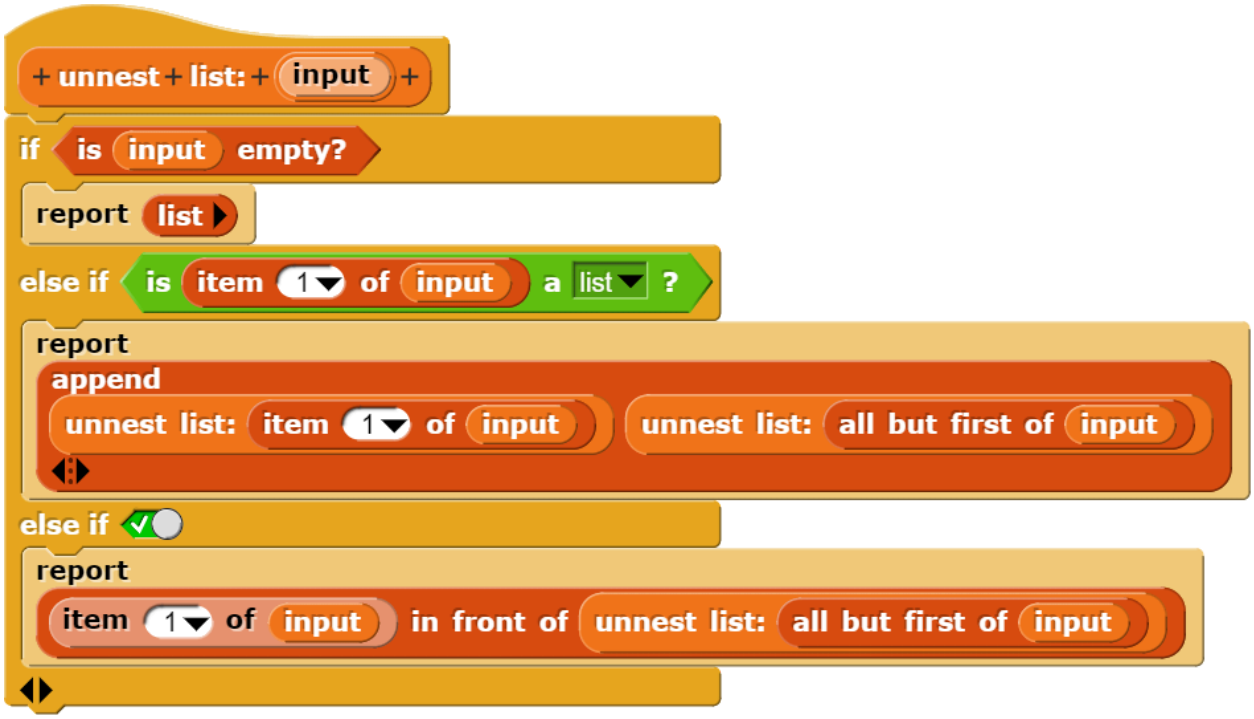
   So, the result is = 3.



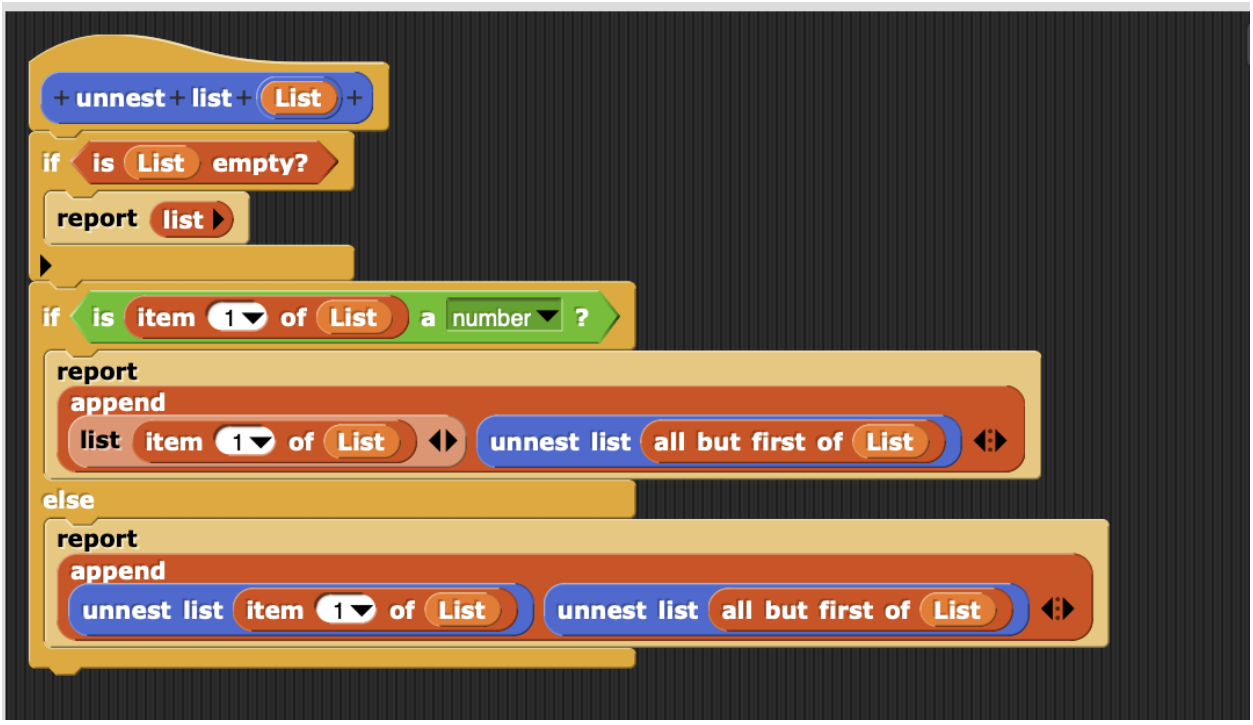
2. Given a very nested list (where we cannot guarantee how nested the input is), write a recursive function that unnests the list to become a 1D list. Here is an example:



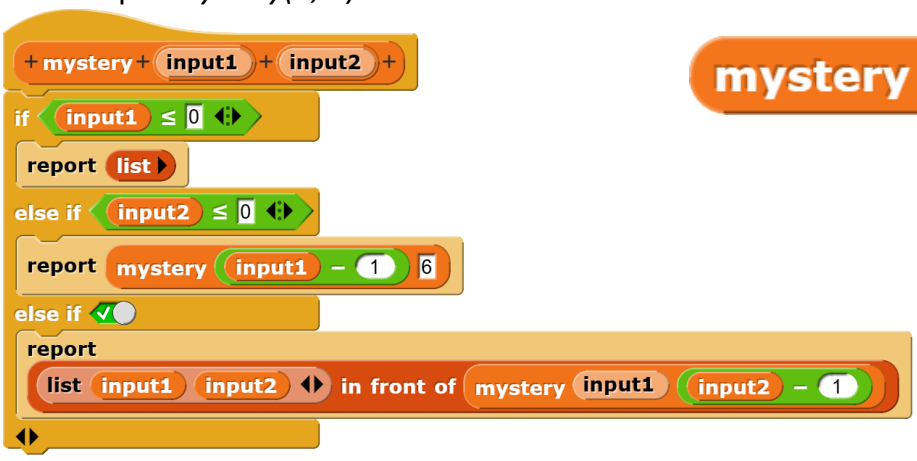
Hint: You may find "append _ _" and "_ in front of _" helpful blocks.



OR (alternate solution)



3. Trace the following function and figure out what the output would be based on the input *mystery(6, 6)*.

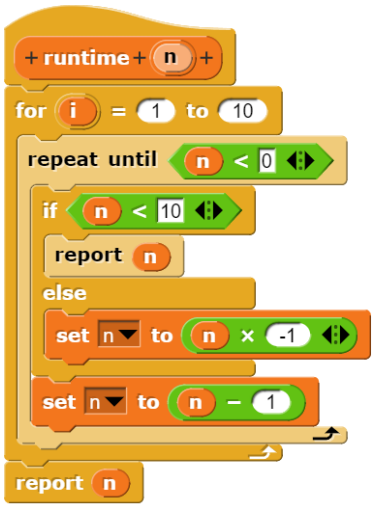




Section II - Algorithmic Complexity

---

1. What is the best case runtime for this function?



   CONSTANT

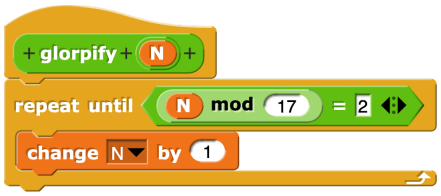
2. What is the worst case runtime for this function?

   CONSTANT

3. If the input, *n*, was 100, what would the function return?

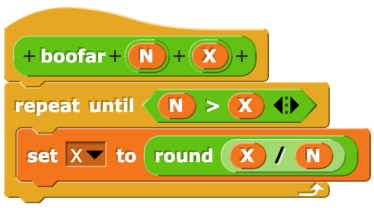
   Returns -101

4. What is the worst case runtime for the following glorpify function?



   CONSTANT

5. *Extra Challenge:* What is the worst case runtime for the following boofar function?



   logX

Optional Section (Extra Practice):

Optional Section I - Tree Recursion

---

1. Related back to the first problem, Hop Hop. Now, we're getting ambitious and decide that we can climb up to K steps at a time. (In the previous problem, K = 2.) Write a function that takes N and K as inputs and returns the number of ways that we can climb the stairs.

2. Write a recursive predicate that checks whether a given item exists in a nested list. The list can contain both values and sublists, and the sublists may also contain sublists. The predicate should return `true` if the item exists (including within nested sublists) and `false` if it does not. Use tree recursion!

Optional Section II - Algorithmic Complexity

---

1. Consider the expression below. If all the helper (non-HOF) blocks operate in constant time, what is the executed runtime of the following script?



2. Assume that the word → list block below executes in linear time as a function of the length of the input word. If myList is a list of n words, each of length n, what is the runtime of the following expression?

