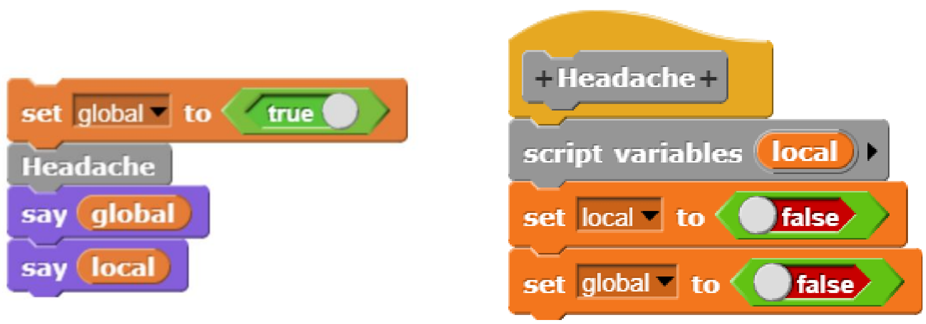
Discussion 4: Scoping, Mutability & Algorithmic Complexity

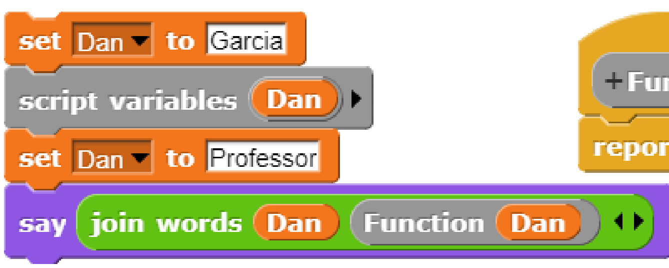
Scoping Practice

For each of the following code snippets, write what the Sprite would say after the script executes. If you believe the code produces any sort of error message, write “Error.” If there are multiple “say” blocks, write the result of each block in a separate box.

a. Assume we create a global variable named “global” (and <u>no other global variables</u>) and then run the script below.



b. Assume we create a global variable named “Dan” and then run the script below.

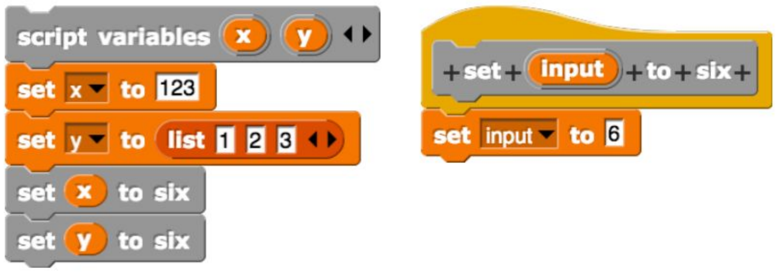


c.

Mutability Practice

What are the values of the script variables x and y after the given script finishes running?

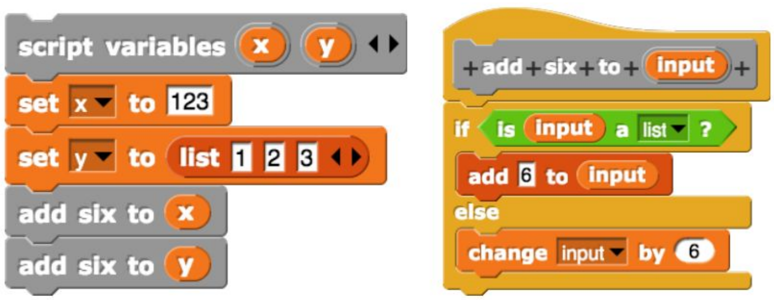
a.



x: ____________

y: ____________

b.



x: ____________

y: ____________

Algorithmic Complexity: Definitions

1. What is runtime? How do we measure it?

____

____

2. If a function runs in O(n) time, that means it runs…

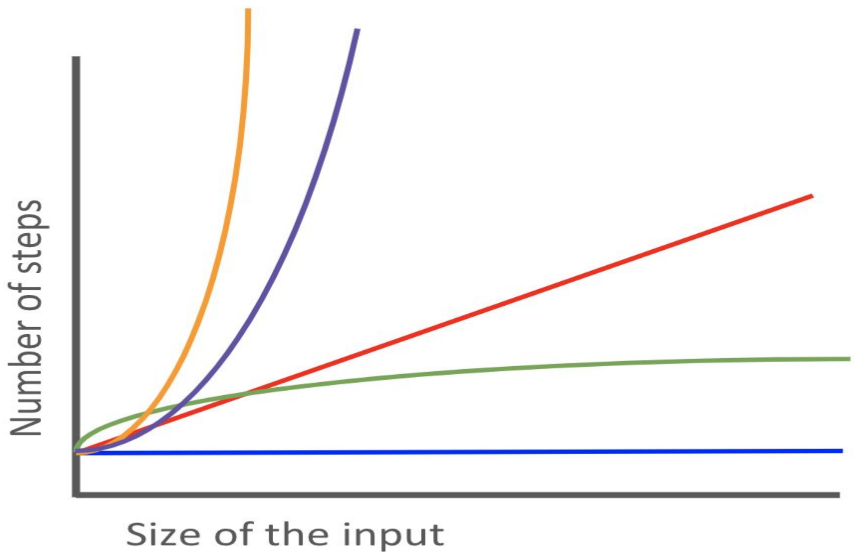
O near time at worst   O near time on average   O near time at best

Understanding Runtimes

1. Fill in the following chart:

Runtime	**Notation**	**As input size increases by…**	**The number of steps change by…**
Constant	C	+1	
Logarithmic (base B)		xB	
Linear		+1	
Quadratic		x2	
Exponential (base B)		+1	

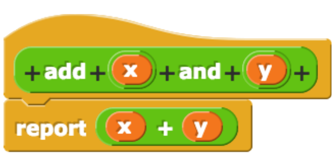
2. In the following diagram, label each of lines. Which is the best runtime? The worst?



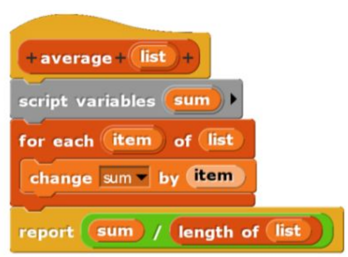
Runtime Practice

1. Find the runtime of each of the following blocks or processes.

a.



b.

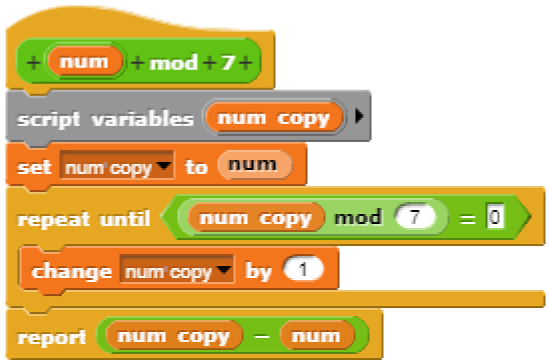


c.

d. This block takes in a value and a list and searches through every item in the list one by one to see if it can find that value.

e. This block takes in a value and a sorted list and searches for the value in the sorted list. Every iteration of the algorithm, it figures out which half of the list the value would be in, and then only searches in that half of the list.

f.



g. You know a secret, and you want to share it with the world. In *state 0*, you are the only person who knows the secret. Then in *state 1*, you share the secret with two friends, so three total people know the secret. Then in *state 2*, both of your friends tell two of their friends, so seven total people know the secret. This pattern (of people sharing the secret with two friends) continues indefinitely. As a function of the *state*, what is the order of growth of the number of people who know the secret?

Challenge Problems

1. What does the following expression do? Assuming that all helper (non-HOF) blocks operate in constant time, what is its runtime?

2. Assume that the word → list block 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?

3. (From Spring 2017 Final Exam) Consider the problem of trying to find out which student has the largest studentID number (SID) in a class of N students, with N being a power of two (2, 4, 8, 16, 32, …). Note that all SIDs are unique. There are four algorithms proposed:

Algorithm I: *The students line up, sitting down. The first two students stand and compare their SIDs. The student with the smaller SID sits back down, the other remains standing. The next student in line stands up and this process repeats until there is only one student (with the largest SID) left standing.*

Algorithm II: *All students stand up, pair up, and simultaneously compare SIDs, and the smaller of each pair sits down. Those still standing repeat the process, pairing up with another standing person, until there is only one left standing; that student has the largest SID.*

Algorithm III: *All students are seated in a circle, facing inward. They write their SID on a sticky note and put it on the back of their neck, number facing out. A random student stands up and walks around to each person and compares his/her number with the number on the neck. If his/hers is larger than all others, he/she declares him/herself as the largest. If they are not, they sit down and someone else (who has not stood up before) stands up, and the process repeats until one person declares he/she is largest.*

Algorithm IV: *Same as Algorithm III but after a person goes around and isn't the largest, rather than the next person being someone who hasn't stood up before, a random person (of the N total students) is chosen again (and it could be someone who has gotten up before).*

Fill in the table for the *worst-case* running time and the *worst-case* number of SID comparisons.

(Select ONE for each algorithm from the top group, and one for each from the bottom).

	Algorithm I	***Algorithm II***	***Algorithm III***	***Algorithm IV***
Constant running time	○	○	○	○
Logarithmic running time	○	○	○	○
Linear running time	○	○	○	○
Quadratic running time	○	○	○	○
Exponential running time	○	○	○	○
May never end, could go forever	○	○	○	○
Constant number of SID comparisons	○	○	○	○
Logarithmic number of SID comparisons	○	○	○	○
Linear number of SID comparisons	○	○	○	○
Quadratic number of SID comparisons	○	○	○	○
Exponential number of SID comparisons	○	○	○	○
Infinite number of SID comparisons	○	○	○	○