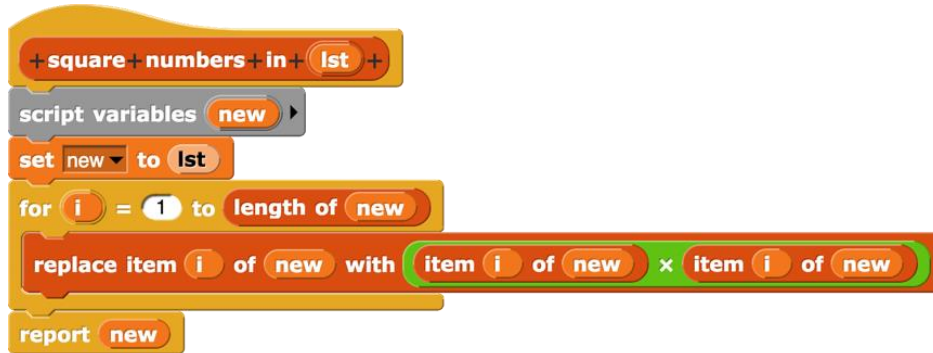


# Discussion 6: Testing & Algorithmic Complexity

## Testing



```
+square+ numbers+ in+ lst +
script variables new
set new to lst
for i = 1 to length of new
  replace item i of new with item i of new x item i of new
report new
```

1. We try to test our code, but we get an error. What does it mean and how can we fix it?



```
test square numbers in w/inputs list 1 2 expecting output
list 1 4
```

Inside: Error  
expecting list but getting number

---

---

## 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...

$O$  in linear time at worst

$O$  in linear time on average

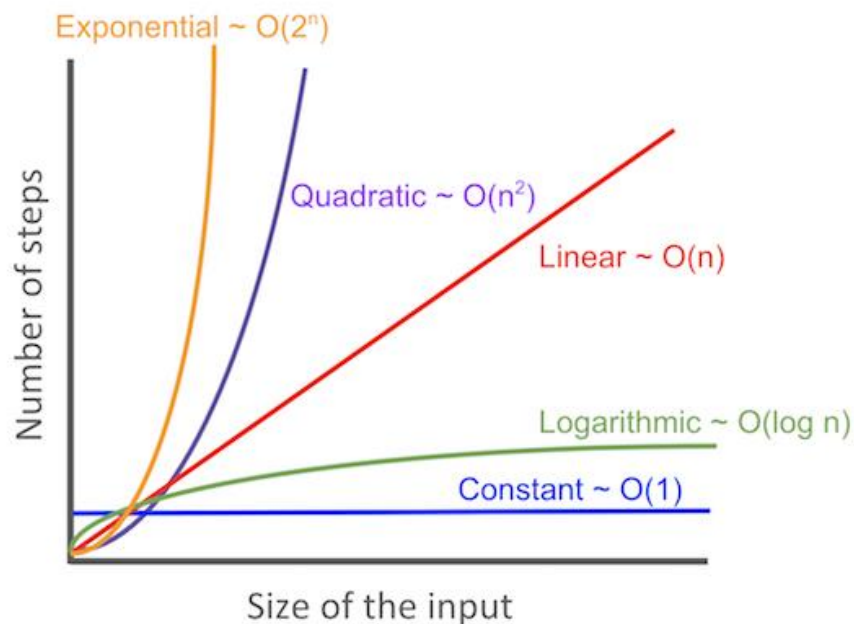
$O$  in linear time at best

## Understanding Runtimes

1. Fill in the following chart:

Runtime	Notation	As input size increases by...	The number of steps change by...
Constant		x2	
Logarithmic (base 2)		x2	
Linear		x2	
Quadratic		x2	
Exponential (base 2)		+1	

2. In the following diagram, which is the best runtime? The worst?



## Runtime Practice

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

a.

```

+add+ x +and+ y +
report x + y
  
```

b.

```

+average+ list +
script variables sum
for each item of list
change sum by item
report sum / length of list
  
```

c.

```

+sort+ list :
script variables sorted list min min_index
set sorted list to list
repeat until length of list = 0
set min_index to 1
set min to item 1 of list
for i = 1 to length of list
if item i of list < min
set min to item i of list
set min_index to i
add item min_index of list to sorted list
delete min_index of list
report sorted list
  
```

d. This process 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 process 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.

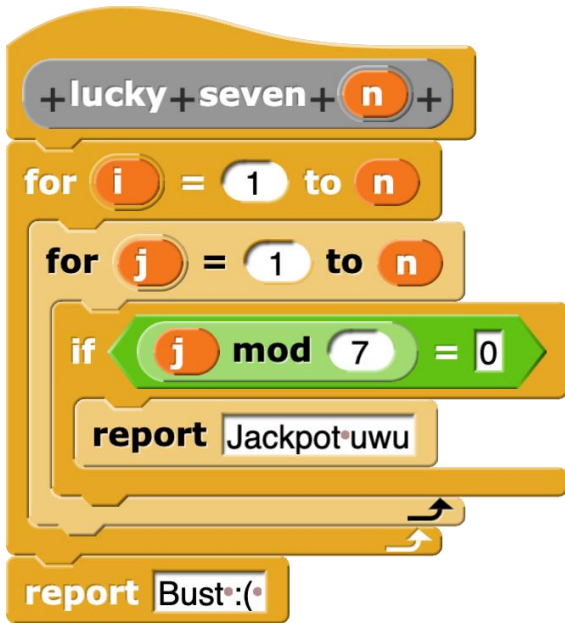
f.

```

+num+ mod+ 7+
script variables num copy
set num copy to num
repeat until num copy mod 7 = 0
change num copy by 1
report num copy - num
  
```

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?

## More Runtime Practice



What is the runtime of this block when  $n$  is less than 7?

What is the runtime of the block when  $n$  is greater than 7?

Why?

```

+ my+func+ input : +
script variables a ▶
set a ▼ to 0
for each item in input
  repeat item
    change a ▼ by 1
report a

```

```

+ my+func+2+ input : +
script variables a ▶
set a ▼ to 0
for i = 1 to length of input
  for i = 1 to length of input
    change a ▼ by 1
report a

```

What do the following calls report? The first one is done for you.

```

my func list 5 5 5 5 5 ◀▶ 25
my func 2 list 5 5 5 5 5 ◀▶
my func list 10 10 10 10 10 ◀▶
my func 2 list 10 10 10 10 10 ◀▶

```

## 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?

