Discussion 4: Algorithmic Complexity + Programming Paradigms

Algorithmic Complexity: Definitions

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

   Runtime measures at what speed an algorithm/block runs. We measure this using the number of steps it takes. Note: We don't use time to measure runtime because computers have different speeds, so the time an algorithm takes to run differs from machine to machine.

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

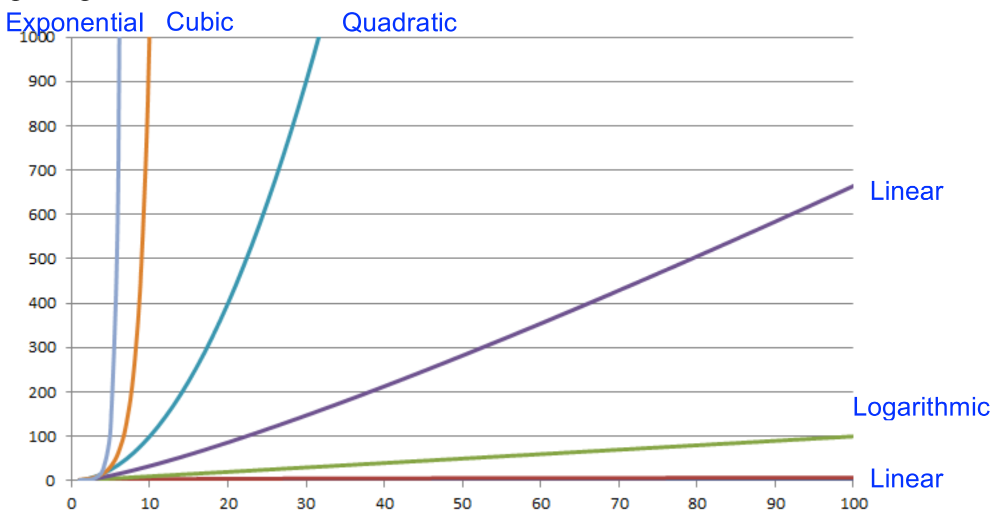
   [✓ O in linear time at worst]   O in linear time on average   O in linear time at best

Understanding Runtimes

1. Fill in the following chart:

Runtime	Notation	As input size increases by…	The number of steps change by…
Constant	O(1)	+1	No change
Logarithmic	O(log n)	x2	+1
Linear	O(n)	+1	+1
Quadratic	O(n^2)	x2	x2
Exponential (base B)	O(B^n)	+1	xB

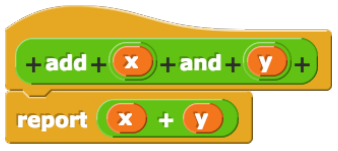
2. In the following diagram, label each of lines. Which is the best runtime? The worst?



Runtime: Practice

1. Find the runtime of the following blocks or descriptions of blocks:

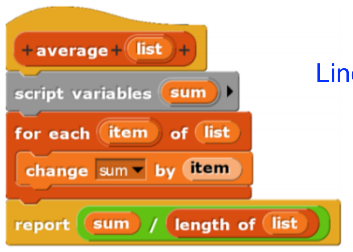
a.



```
+add+ (x) +and+ (y) +
report ((x) + (y))
```

Constant

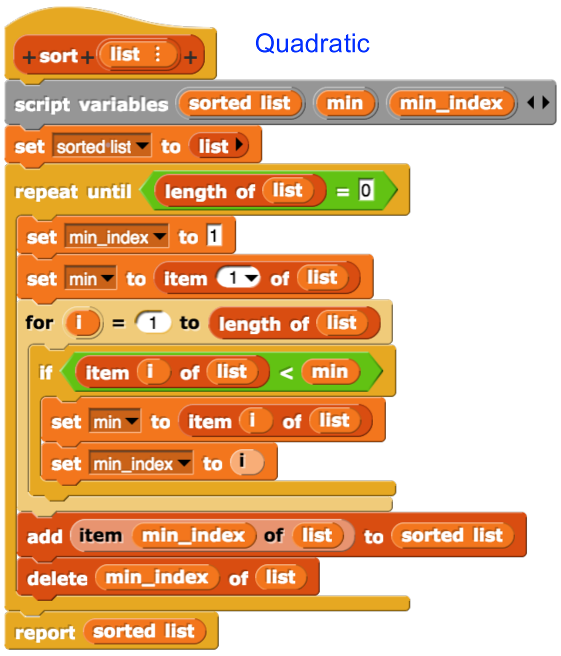
b.



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

Linear

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)
```

Quadratic

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.

Linear

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.

Logarithmic (this is binary search, or find number in sorted list) from lab!

Programming Paradigms

1. Write down the programming paradigm that **best** fits the following descriptions:

a. One sprite tells a second sprite to run some code. The second sprite does it. Object Oriented

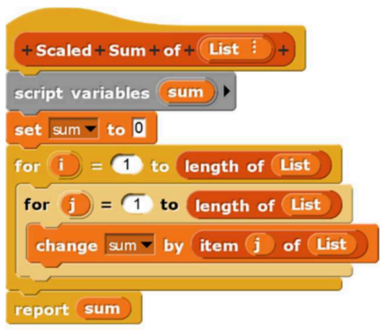
b. You input a global list into a block. It reports a new list with different values, without modifying the input list. Functional

c. You give a program a condition as an input and it uses this condition to remove numbers from a list. You input a list and it removes items. Declarative

d. You have a global variable set to a secret word. You change the secret word every time you ask a player for a new secret word. Imperative

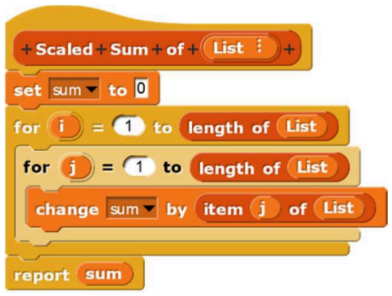
2. Match each of the following scripts to a programming paradigm:

Functional ________ Imperative ________ Object Oriented ________

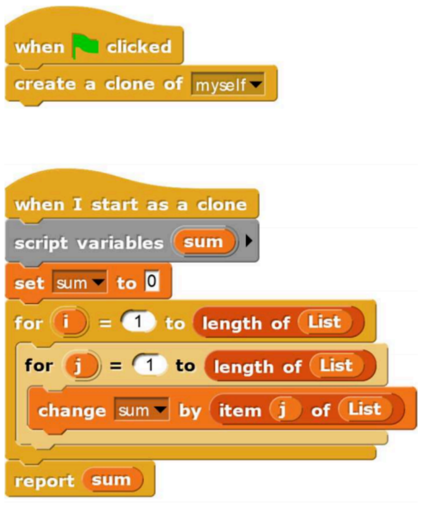


```
+Scaled+Sum+of+ (List :) +
script variables (sum)
set sum to 0
for (i) = 1 to (length of (List))
    for (j) = 1 to (length of (List))
        change sum by (item (j) of (List))
report (sum)
```

Assume `sum` *is a global variable in the second script*



```
+Scaled+Sum+of+ (List :) +
set sum to 0
for (i) = 1 to (length of (List))
    for (j) = 1 to (length of (List))
        change sum by (item (j) of (List))
report (sum)
```

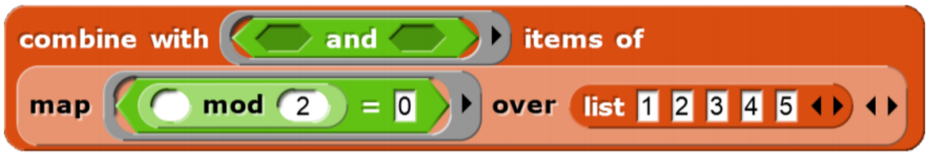


```
when green flag clicked
create a clone of myself

when I start as a clone
script variables (sum)
set sum to 0
for (i) = 1 to (length of (List))
    for (j) = 1 to (length of (List))
        change sum by (item (j) of (List))
report (sum)
```

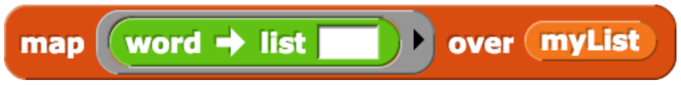
Challenge

1. What does the following block do? What is its runtime?



This block reports true if there are only even numbers in the input list. Since HOFs take linear time, and the mod and equals blocks take constant time, this runs in linear time.

2. If myList is a list of n words, each of length n, what is the runtime of the following block?



Note: We will tell you when a block runs in linear time.

In this case, word -> list runs in linear time with respect to the length of the word. Since myList has n elements, map will do n actions, one for each element. For each element, word -> list takes n time, so the total runtime is O(n^2).