Big O Cheat Sheet:

-Big Os-

O(1) Constant- no loops

# O(log N) logarithmic- usually searching algorithms have log(n) if they are sorted (Binary Search) (not on hash

O(n) Linear- for loops while loops

O(n^2) Quadratic- every element in a collection needs to be compared to ever other element. Two nested loops

O(n!) Factorial- you are adding a loop for every element.

\*\*Iterating through half a collection is still O(n)

\*\*Two separate collections: O(a \* b)

-Rule Book-

Rule 1: Always worst case

Rule 2: Remove Constants

Rule 3: Different inputs should have different variables. O(a+b). A and B arrays nested would be O(a\*b)

+ for steps in order

\* for nested steps

Rule 4: Drop Non-dominant terms

-What can cause time in a function?-

Operation (+, -, \*, /)

Comparisons (<, >, ==)

Looping (for, while)

Outside Function call (function())

What is good code?

1. Readable
2. Scalable (speed, memory)

Which code is best?

Readable, Speed (Time Complexity), Memory (Space Complexity)

-What causes Space complexity?

Variables

Data Structures

Function Call

Allocations