

# Week 2



- Login
- Open Python
- wait for further instruction

# Class Structure

- Last week explanation
- Introduce algorithm complexity
- Reveal this weeks questions



# 5/12/2019 Rankings

1	 695	toxekgfx	79	25
2		ktissan	55	21
3		593368	21	3
4		Parneet_J14	3	1
4		Bhagya_K	3	1
4		DeronB	3	1
4		Sohban_M	3	1
4		Aj	3	1
4		10deep_	3	1
4		dhanush2003	3	1
4		Amsan5941	3	1
4		sajeenth19	3	1

# Pseudo Code for J2 Shifty Sum

1. Take inputs (number and shift value)
2. Create counter = 0
3. Create result variable to store the answer result = 0
4. Loop from 0 to shift value+1 (Since we want to shift n by k amounts)
  - a.  $\text{result} += (\text{number} * 10^{\text{counter}})$   $(12*10^0) + (12*10^1) + (12*10^2) \dots$

Print out result

Remember: that the range(n) function starts at 0, and goes to n-1

(For example range(5) = 0,1,2,3,4 )

# Pseudo Code for S2 Pretty Average Primes

1. Create a function to find if a number is prime or not.
  - a. Loops from 2- $\sqrt{\text{number}}$
2. Loop through from 0 to T: (T represents each integer)
  - a. Check if number is prime and if  $(2 \cdot T)$ -number is prime
    - i. Store both numbers in a list
    - ii. Break loop
3. Print out all the numbers

# Prime efficient algorithm

```
import math

def test_prime(n):
    for i in range(2, round(math.sqrt(n))):
        if n % i == 0:
            return False
    return True
```

- In this method we only check numbers from 2 to  $\sqrt{n}$
- Remember to copy this code down and implement it into yours!

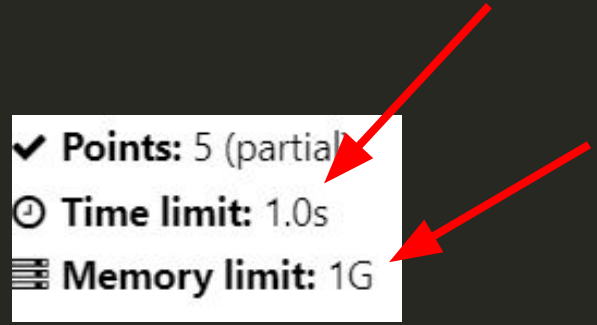
# Algorithm Complexity

The **time complexity** of a program is the amount of time it takes to run for all inputs

The **space complexity** of a program is the amount of space it takes in bytes

## Why does this matter?

- We are given a time and space limit of all our programs.
- If we exceed the limit we will fail the problem



✓ **Points:** 5 (partial)  
🕒 **Time limit:** 1.0s  
📄 **Memory limit:** 1G

# Time Complexity

- If a program requires the same amount of time to run on all inputs, then the program has a constant time complexity.

e.g)

```
first_number = int(input())  
second_number = int(input())  
  
print(first_number+second_number)
```



# Time Complexity

- If the amount of time of a algorithm increases with the increase of input value then it has a linear time complexity.

e.g)

```
n = int(input())  
for i in range(n):  
    print(i)
```

time complexity can be  
represented by  $O(n)$

# Exercise

Find the time complexity of this program:

```
n = int(input())  
  
for i in range(n):  
    for x in range(n):
```

# Quadratic time complexity

- The running time of a nested loop is proportional to the square of  $n$

Therefore the time complexity is  $n^2$

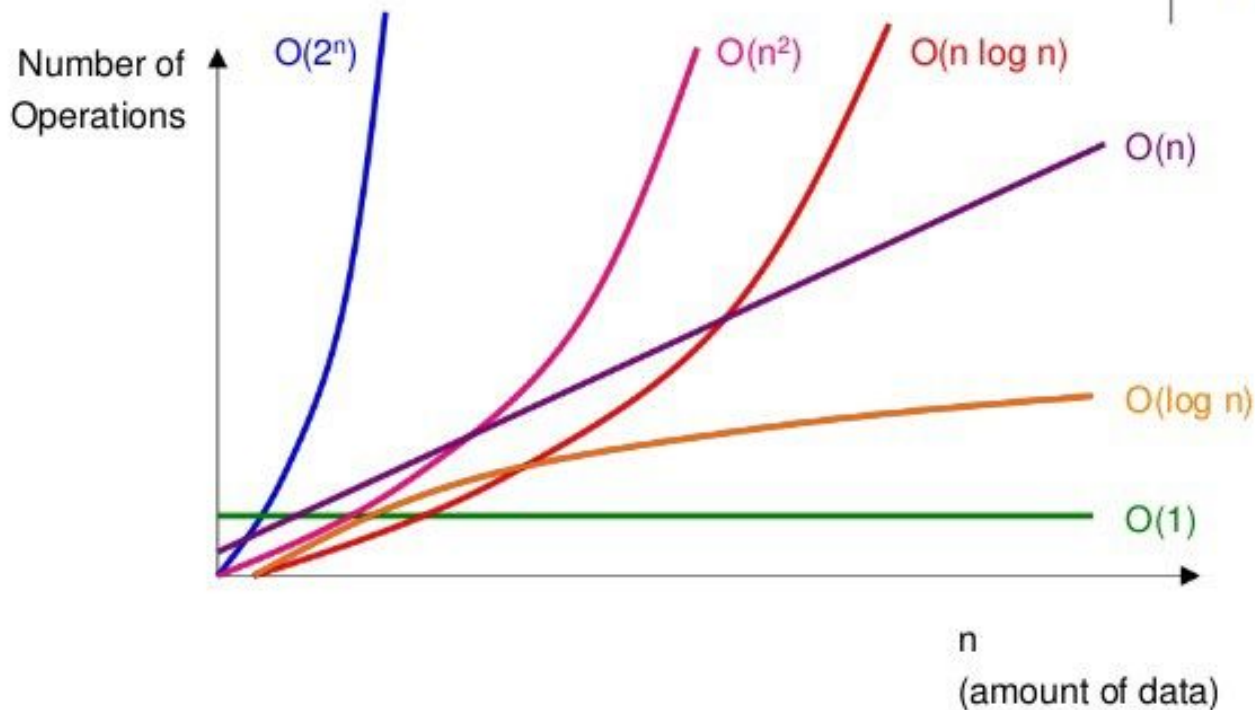
# Exercise

Find the time complexity of this program:

```
import math

def test_prime(n):
    for i in range(2, round(math.sqrt(n))):
        if n % i == 0:
            return False
    return True
```

# Comparing Big O Functions



# This weeks problems

## Grade 11's:

2016 J3 - Hidden Palindrome

2012 J4 - Big Bang Secrets

## Grade 12's:

2017 S1 - Sum Game

2017 S2 - High Tide, Low Tide

\*move on to the next problem one only after finishing the first problem

For beginner programmers go to [learnpython.org](https://www.learnpython.org/) and try doing 'Hello World' tutorial