

# Discussion 12

**Final Review; Final Discussion** 🙏

Antonio Kam

`anto [at] berkeley [dot] edu`

**All slides can be found on**

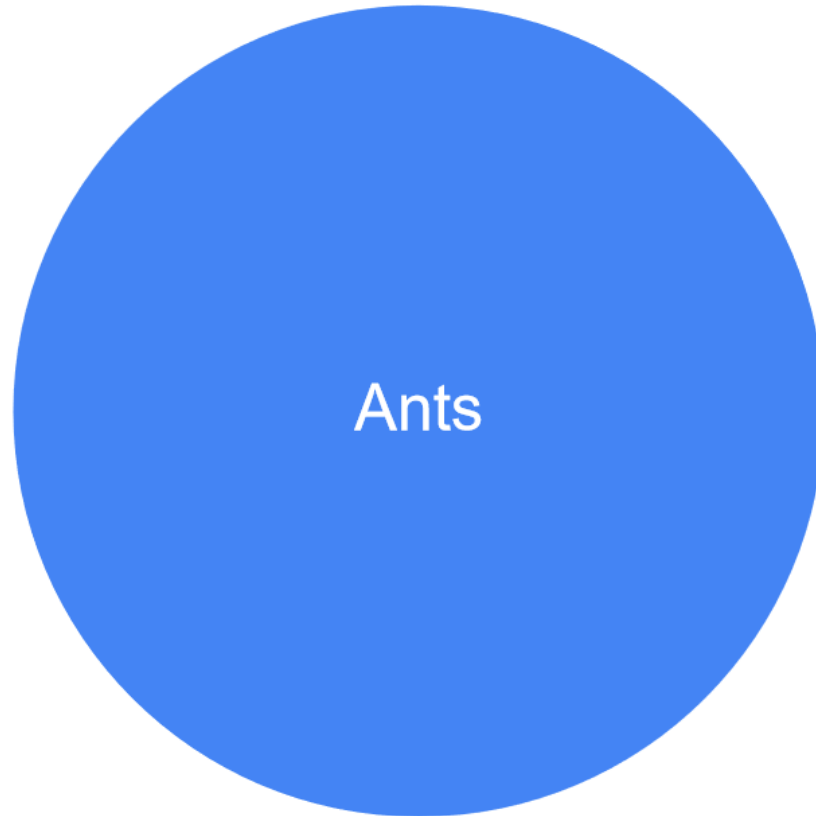
**[teaching.roux1.es](http://teaching.roux1.es)**

# Announcements

- This is the final discussion 😭
  - There's one more section (lab! - final review please come; results from this discussion will be shown then)
- Special Topics lectures are in scope!
  - You will be asked surface-level questions on them (basically if you watch the lecture you should be fine)
- Scheme
  - Checkpoint 2 due tomorrow!
  - Final due on Tuesday
  - Please finish these early 🙏
- HW 07, HW 08

# Results from last section

Do you like Ants or Scheme more?



# Notes from last section

- Will there be review sessions before final?
  - Yes! (I'm doing the one for Recursion and Tree Recursion - schedule will be released on Piazza in the future)
  - For now it's set 12:30 PM to 2:00 PM in Soda 271
- "It's not you, it's RegEx"
- The shirt I wear
- Favorite game
  - Ori!
  - PMD
- Monkeypox 🙄
  - Imao i wish i knew

# Temperature Check/Vote

- Recursion
- Mutation (Lists)
- Efficiency (I will cover this)
- Trees
- Linked Lists
- Scheme
- RegEx

# Efficiency

Order of growth	Description
Constant	Always the same number of steps regardless of the input size
Logarithmic Growth	Number of steps increases proportionally to the logarithm of the input size
Linear Growth	Number of steps increases in direct proportion to the input size
Quadratic Growth	Number of steps increases in proportion to the square of the input size
Exponential Growth	Number of steps increases faster than a polynomial function of the input size

# Constant Growth

```
def constant1(n):  
    print(n)  
    return  
  
def constant2(n):  
    for _ in range(100):  
        print(n)  
    return
```



# Logarithmic Growth

```
def log1(n):  
    while n > 0:  
        print(n)  
        n = n // 2 # based on log_2
```

```
def log2(n):  
    if n == 0:  
        return 1  
    elif n % 2 == 0:  
        return (log2(n // 2)) ** 2  
    else:  
        return n * log2(n - 1)
```

# Linear Growth

```
def linear1(n):  
    while n > 0:  
        print(n)  
        n -= 1
```

```
def linear2(lst):  
    for x in lst:  
        print(x)
```

# Quadratic Growth

```
def linear1(lst):  
    for row in lst:  
        for item in lst:  
            print(item)
```

# Exponential Growth

```
def exp1(n):  
    if n == 0 or n == 1:  
        return n  
    else:  
        return exp1(n - 1) + exp1(n - 2)
```

# Worksheet!

# Attendance

[links.roux1.es/disc](https://links.roux1.es/disc)

# Mental Health Resources

- CAPS:
  - If you need to talk to a professional, please call CAPS at 510-642-9494.
- After Hours Assistance
  - For any assistance after hours, details on what to do can be found at [this link](#)

# Anonymous Feedback Form

[links.roux1.es/feedback](https://links.roux1.es/feedback)

Thanks for coming! 🎉

*Please give me feedback on what to improve!*