Net Neutrality and Functions Part 3

In this lecture, we will cover

- Net Neutrality - and other "laws" governing the internet
- More options for function parameters
  - Default parameters
  - named parameters
  - functions as parameters

Your future in CS

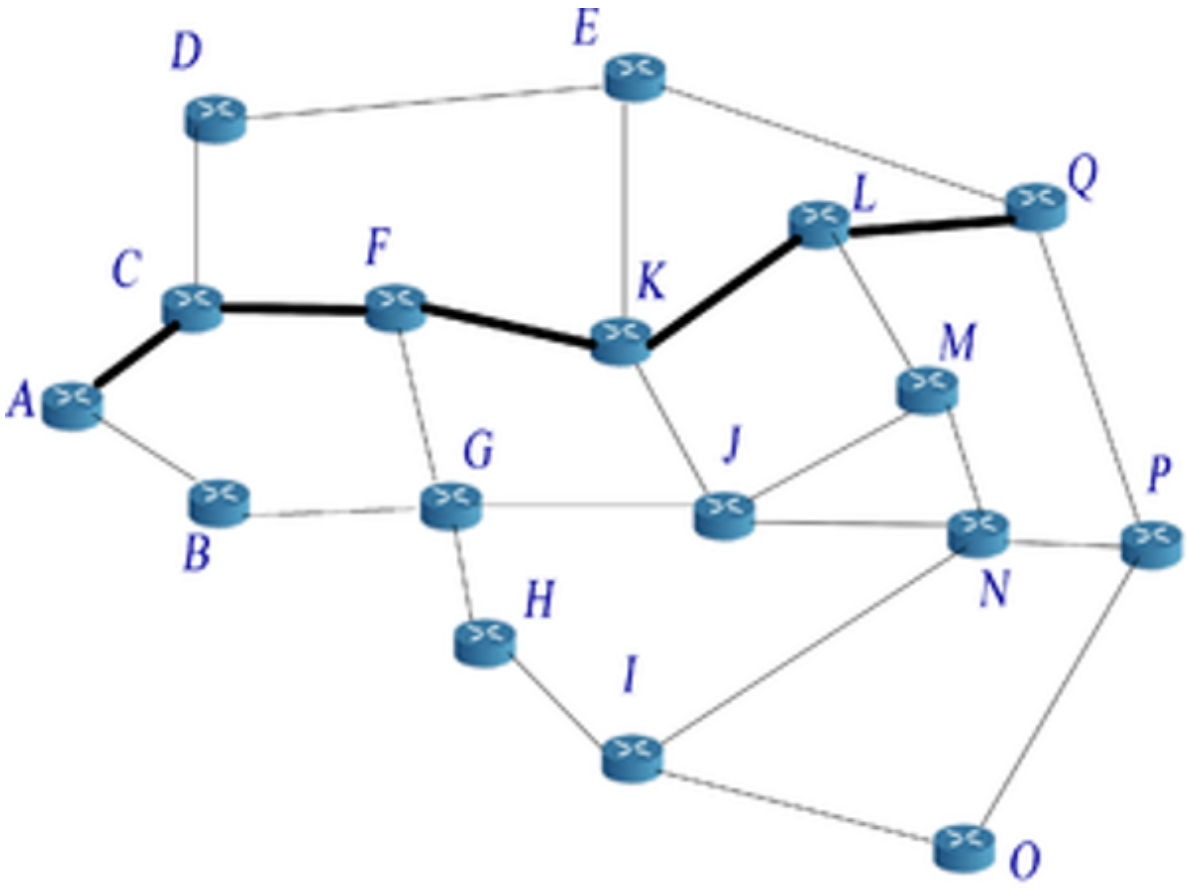
I used to include this on my slides, but since these slides have changed - going to just leave it up here for every notebook. I get a lot of questions about more programming courses, the concentrations, and minors in computer science. Here is a brief reminder.

CS 164 – Next Course In Sequence, also consider CS 220 (math and stats especially)

- CO Jobs Report 2021 – 77% of *all* new jobs in Colorado require programming
- 60% of all STEM jobs requires *advanced* (200-300 level)
- 31% of all Bachelor of Arts degree titled jobs also required coding skills
- 2016 Report found on average jobs that require coding skills paid $22,000 more

- Concentrations in CS:

  - Computer science has a number of concentrations.
    - General concentration is the most flexible, and even allows students to double major or minor pretty easily.
    - Software Engineering
    - Computing Systems
    - Human Centered Computing
    - Networks and Security
    - Artificial Intelligence
    - Computer Science Education.
  - Minors:
    - Minor in Computer Science - choose your own adventure minor
    - Minor in Machine Learning - popular with stats/math, and engineering
    - Minor in Bioinformatics - Biology + Computer Science



- Computer *Mesh* networks
  - Essential for how the internet works
  - Computer A connects to computer Q
    - Information is broken up into packets
    - Follows multiple routes
    - Reassembled at the destination
- Thinking about 'packets' more in depth:
  - Header:
    - length of packet
    - synchronization (bits that help match up the various packets)
    - packet number
    - protocol
      - email, webpage, streaming video, etc
    - destination address
    - originating address
  - Payload
    - Body / data itself
  - Footer / Trailer
    - Error checking data
- Differentiated Services
  - Since we know:
    - protocol
    - destination address

        - originating address
    - We can pick and choose which packets have priority
        - Does voice demand higher priority than video games?
            - Ok, what about emergency services?

Differentiated Services - Net Neutrality Debate

- Net Neutrality "simplest form"
    - All packets should be allowed to reach their destination
        - Without speed delays based on origin, destination, or protocol
- FCC (Federal Communications Commission)
    - Regulates the flow of network information
    - Based on an issue with telephones
        - Company paid to have all calls for their competitor routed to their number
- Problems
    - Telephone lines are centralized services
        - The 'imporant' part of phone lines exist at the phone company.
    - Networks are decentralized services
        - Routing and 'intelligent' systems happen at / near the clients!
        - There isn't a single routing point for anything!
- Type of traffic between sources
    - Voice > video > games, etc
        - Many major companies such as Google and Verizon agree this is valid
        - Government supports this idea
        - But what if we then make it a pay based model?
            - One company pays to have their data go faster than another?
            - This ensures important information gains priority?
            - Right?
        - Even more extreme, what if companies compete to control data?
            - TELUS - 2005 - blocked sites supporting labor strike against company (along with 766 unrelated sites)
            - AT&T, SPRINT and VERIZON: From 2011–2013 - blocked google wallet
            - EUROPE: A 2012 report - 1 in 5 clients affected by slowdowns due to blocked packets and sites

Discussion

- Build a list of Pros and Cons for **AND** against net neutrality
    - You are free to pull up arguments from online
    - Actually, you are encouraged to find the 'standard arguments'
- Tables will be randomly called upon to present one

The next part, let's use our understanding of the technology to see these pro/cons we often see.

Other Internet Laws?

- Taxes - and trading across state lines
- What about international sales?
- Where are the companies based?
- Whose laws do you follow on what can be sold
  - Think about fireworks or editables
- Where do you stand on technology and the laws around it?
- The internet will always have disruptive tech
  - Is keeping up a losing battle?
  - Also, this is one thing that makes computer science so interesting - we are constantly changing the world in unexpected ways!

Call to Action

- There is a desperate need for lawyers and policy makers that truly understand technology
  - We have students earning Majors + the legal minor for this reason
  - We also have philosophy Majors + CS minors who are planning to apply to law school
    - Patent law, but also policy makers are political science + CS
- CS-164 is the next CS course
- If you are still in this class, you will do fine in CS 164

Python Functions Part 3

- Reminder
  - A function is a block of code with a set purpose
  - That can be repeatedly called

In Class Activity

Write a function that takes in the following parameters:

- start
- end
- step

And builds a list based on that range, inclusive of start, exclusive of end with a step of step. For example:

```
number_generator(0, 10, 1)  # returns  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
number_generator(0, 10, 2)  # returns [0, 2, 4, 6, 8]
```

In [ ]:
```
def number_generator(start, end, step):
    lst = []
    while start < end:
        lst.append(start)
        start += step
```

```
    return lst


print(number_generator(0, 10, 1))  # returns  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(number_generator(0, 10, 2))  # returns  [0, 2, 4, 6, 8]
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 2, 4, 6, 8]
```

However, what if we wanted certain parameters to 'default' to other values?

In [ ]:
```
def number_generator(start, end = 10 , step=1):
    lst = []
    while start < end:
        lst.append(start)
        start += step
    return lst
```

End and Step now become optional - putting in the default values

In [ ]:
```
print(number_generator(0))
print(number_generator(0, 5))
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 1, 2, 3, 4]
```

We can also call specific parameters

In [ ]:
```
print(number_generator(0, step=2))
print(number_generator(start=5, end=7, step = 1))
```

```
[0, 2, 4, 6, 8]
[5, 6]
```

Overall, just more features we can use with function parameter to make our easier! Especially helpful with default values.

We can also allow a variable number of arguments called an 'args list'. print does this!

Note: it has to be the last argument in any parameter!

In [ ]:
```
def print_hello(*args):
    for arg in sorted(args):
        print(f"Hello {arg}, ", end="")
    print()

print_hello("Wendy", "Hook", "John", "Peter")
print_hello("Wendy", "John")
```

```
Hello Hook, Hello John, Hello Peter, Hello Wendy,
Hello John, Hello Wendy,
```

Functions as Parameters

- A major component of functional languages is that functions are 'first order' objects.
  - They can be used as any other value

- Including setting functions to variables / parameters

In [ ]:
```
## recall
values = [10, 3, 4, 8, 9, 10]
max_val = max(values)
print(max_val)

min_val = min(values)
```

```
10
```

What if we wanted to build a function that prints out the 'winner' of the values

In [ ]:
```
def print_winner(values, win_condition):
    print("The winner is", win_condition(values))


print_winner(values, max)
print_winner(values, min)
```

```
The winner is  10
The winner is  3
```

We can also use default values, which is common!

In [ ]:
```
def print_winner(values, win_condition=max):
    print("The winner is", win_condition(values))

print_winner(values)
print_winner(values, min)
print_winner(values, sum) # just to show something else
```

```
The winner is 10
The winner is 3
The winner is 44
```

Overall

- Functions have a lot of power in python
- While you don't need default parameters, they often make a lot of sense - so use them!
- Also, passing functions in parameters helps make some code 'more elegant' / versatile (often used in machine learning)