[320] Web 4: Web Crawling

Tyler Caraza-Harter

Review requests, flask, selenium

Which module is for writing servers?

1. requests
2. flask
3. selenium

If you need to JavaScript engine to load data on the page, you should scrape with

1. requests
2. selenium

The DOM (Document Object Model) is an example of a:

1. DAG
2. Tree
3. Binary Tree

T/F: if `b` is a selenium browser visiting a static page, then `b.page_source` gives the HTML of that .html file as a string

1. True
2. False

Review requests, flask, selenium

Which module is for writing servers?

1. requests
[✓ 2. flask]
3. selenium

If you need to JavaScript engine to load data on the page, you should scrape with

1. requests
[✓ 2. selenium]

The DOM (Document Object Model) is an example of a:

1. DAG ← True, but not most informative answer...
[✓ 2. Tree]
3. Binary Tree

T/F: if `b` is a selenium browser visiting a static page, then `b.page_source` gives the HTML of that .html file as a string

1. True
[✓ 2. False] ← It is a translation of DOM (which may have changed) back to HTML

Internet and Graphs

Demo: Breadth-First Search on Pages

A.html

Welcome!
Please visit page B

C.html

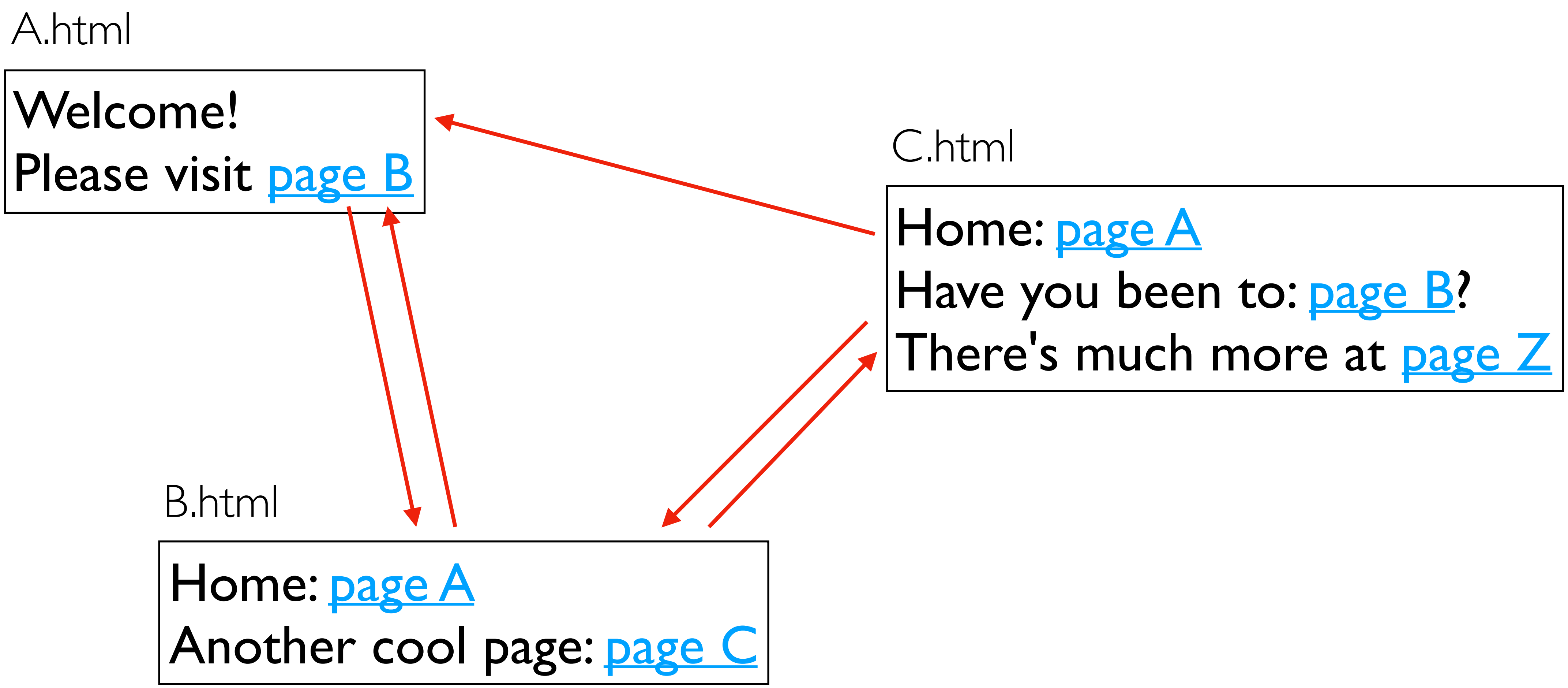
Home: page A
Have you been to: page B?
There's much more at page Z

B.html

Home: page A
Another cool page: page C

pages are **nodes**

Demo: Breadth-First Search on Pages



A.html

Welcome!

Please visit page B

C.html

Home: page A

Have you been to: page B?

There's much more at page Z

B.html

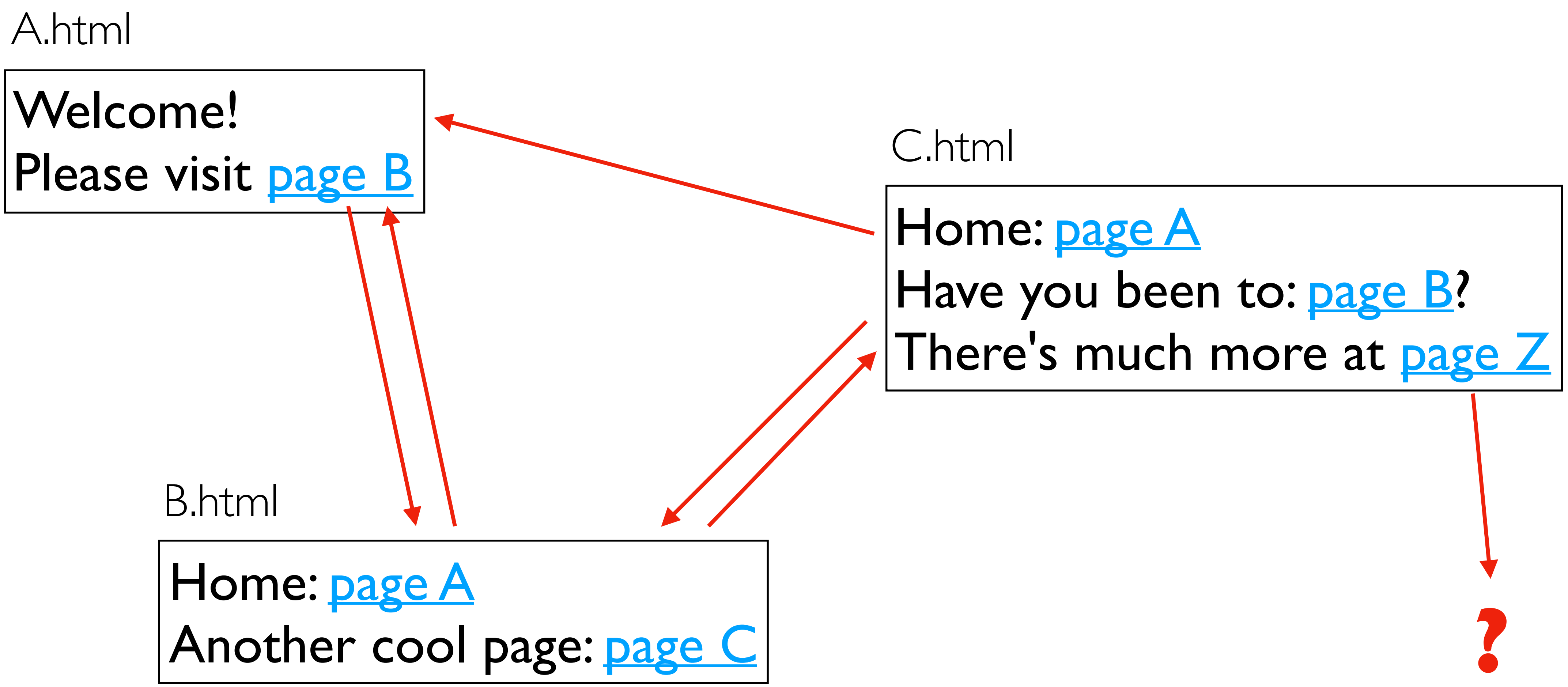
Home: page A

Another cool page: page C

pages are **nodes**

links are **edges**

Demo: Breadth-First Search on Pages



A.html

Welcome!
Please visit page B

C.html

Home: page A
Have you been to: page B?
There's much more at page Z

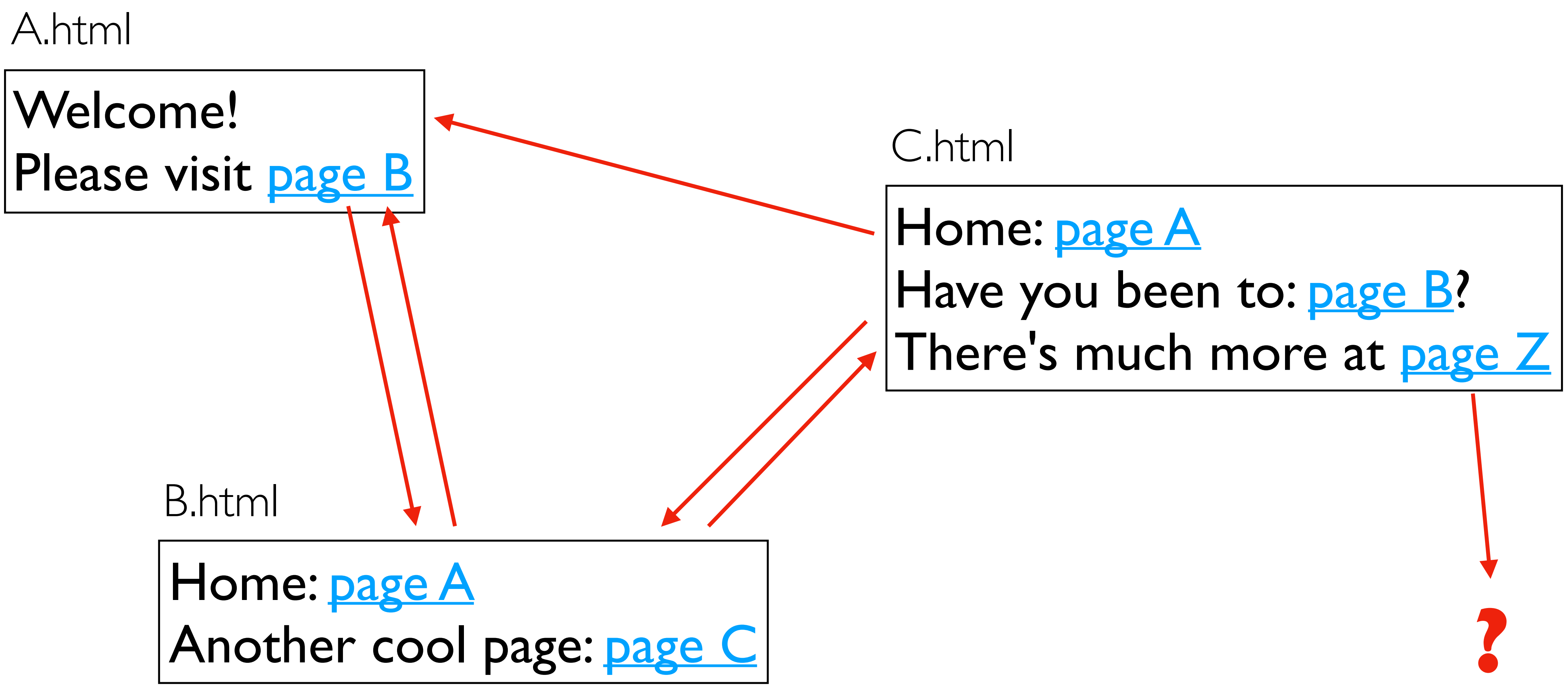
B.html

Home: page A
Another cool page: page C

?

pages are **nodes**
links are **edges**
we might do BFS/DFS without ever seeing whole graph

Demo: Breadth-First Search on Pages



A.html

Welcome!
Please visit page B

C.html

Home: page A
Have you been to: page B?
There's much more at page Z

B.html

Home: page A
Another cool page: page C

Practice: https://tyler.caraza-harter.com/cs320/s20/lectures/lec-20/practice.html

HTTP 429: Too Many Requests

```python
import time
from flask import request, Flask, Response

app = Flask(__name__)

last_req = {}

def backoff(fn):
    def wrap():
        t0 = last_req.get(request.remote_addr, 0)
        t1 = time.time()

        if t1 - t0 < 2:
            r = Response("<h1>backoff! 429</h1>")
            r.status_code = 429
            return r

        last_req[request.remote_addr] = t1
        return fn()
    wrap.__name__ = fn.__name__
    return wrap

@app.route("/")
@backoff
def home():
    return "<h1>Hello!</h1>"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="5003")
```

no IP address is allowed to send us
more than 2 requests per second

```python
import time
from flask import request, Flask, Response

app = Flask(__name__)

last_req = {}

def backoff(fn):
    def wrap():
        t0 = last_req.get(request.remote_addr, 0)
        t1 = time.time()

        if t1 - t0 < 2:
            r = Response("<h1>backoff! 429</h1>")
            r.status_code = 429
            return r

        last_req[request.remote_addr] = t1
        return fn()
    wrap.__name__ = fn.__name__
    return wrap

@app.route("/")
@backoff
def home():
    return "<h1>Hello!</h1>"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="5003")
```

no IP address is allowed to send us more than 2 requests per second

```python
r.headers["Retry-After"] = 2 - (t1 - t0)
```

tell clients when they're welcome again

robots.txt

Examples

https://reddit.com/robots.txt

https://en.wikipedia.org/robots.txt

https://news.ycombinator.com/robots.txt

https://docs.python.org/3/library/urllib.robotparser.html

```
from urllib.robotparser import RobotFileParser
robo = RobotFileParser("https://news.ycombinator.com/robots.txt")
robo.read()
robo.crawl_delay(useragent="320-agent")
```

one visit per 30 seconds

```
robo = RobotFileParser("https://tyler.caraza-harter.com/robots.txt")
robo.read()
base = "https://tyler.caraza-harter.com/cs320/s20/lectures/lec-20/calendar/"
print(robo.can_fetch("320-agent", base+"A.html"))
print(robo.can_fetch("320-agent", base+"months/JAN-2020.html"))
```

why forbidden?

Demo: Crawling with robots.txt

https://tyler.caraza-harter.com/cs320/s20/lectures/lec-20/calendar/A.html

```
robo = RobotFileParser("https://tyler.caraza-harter.com/robots.txt")
robo.read()
base = "https://tyler.caraza-harter.com/cs320/s20/lectures/lec-20/calendar/"
print(robo.can_fetch("320-agent", base+"A.html"))
print(robo.can_fetch("320-agent", base+"months/JAN-2020.html"))
```

why forbidden?