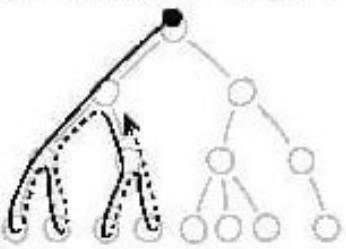


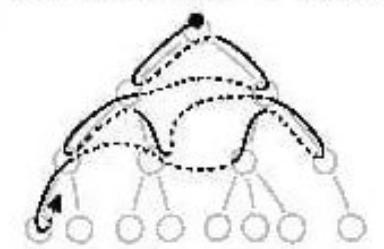
Welcome!

GM #13? | HHS CS

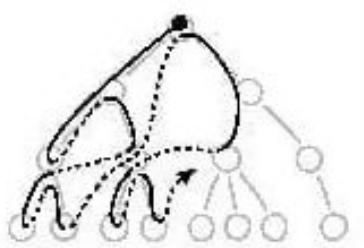
DEPTH-FIRST SEARCH



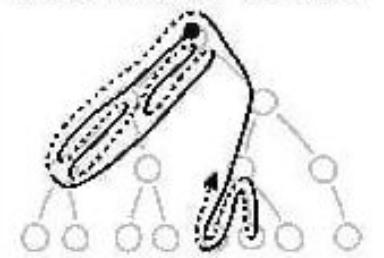
BREADTH-FIRST SEARCH



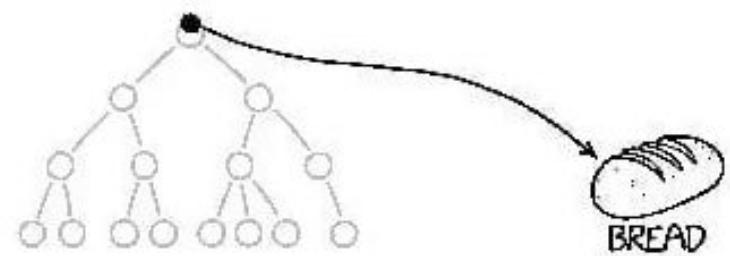
BREPTH-FIRST SEARCH



DEADTH-FIRST SEARCH



BREAD-FIRST SEARCH



> Wikipedia Game

Goal: To get from one Wiki page to another by clicking links
(ideally the least number)

Try:

Macadamia

<https://en.wikipedia.org/wiki/Macadamia>

To

Computer science

https://en.wikipedia.org/wiki/Computer_science

> Solution (one of)

Macadamia: <https://en.wikipedia.org/wiki/Macadamia>

Genus: <https://en.wikipedia.org/wiki/Genus>

Taxonomic_rank: https://en.wikipedia.org/wiki/Taxonomic_rank

Hierarchy: <https://en.wikipedia.org/wiki/Hierarchy>

Computer_science:

https://en.wikipedia.org/wiki/Computer_science

> How does the code work?

hhscs.club

Searches through Wikipedia page for links

Python Libraries for Web Scraping

- Beautiful Soup
- Re (regex)

> Website HTML file (inspect element)

Paper		163 languages			
Article	Talk	Read	View source	View history	Tools
From Wikipedia, the free encyclopedia					
<p>For other uses, see Paper (disambiguation).</p>					
<p>Paper is a thin sheet material produced by mechanically or chemically processing cellulose fibres derived from wood, rags, grasses, or other vegetable sources in water, draining the water through a fine mesh leaving the fibre evenly distributed on the surface, followed by pressing and drying. Although paper was originally made in single sheets by hand, almost all is now made on large machines—some making reels 10 metres wide, running at 2,000 metres per minute and up to 600,000 tonnes a year. It is a versatile material with many uses, including printing, painting, graphics, signage, design, packaging, decorating, writing, and cleaning. It may also be used as filter paper, wallpaper, book endpaper, conservation paper, laminated worktops, toilet tissue, currency, and security paper, or in a number of industrial and construction processes.</p>		<p>Paper</p>		<p>49.79 x 15.43</p>	
Material type	Thin material	Physical properties		Density (ρ)	From 10 gsm to 3000 gsm
History		Traditional Chinese	Simplified Chinese	History	
See also	References	External links		Help	

```
Elements Console Recorder ▾ Performance insights ▾ >   

79718">  
  <link rel="mw-deduplicated-inline-style" href="mw-data:TemplateStyles:r10664  
79718">  
  <link rel="mw-deduplicated-inline-style" href="mw-data:TemplateStyles:r10664  
79718">  
▶ <table class="infobox">(...)</table>  
▼ <p>  
  <b>Paper</b>  
  " is a thin sheet "  
  ...  
  <a href="/wiki/Material" title="Material">material</a> == $0  
  " produced by mechanically or chemically processing "  
  <a href="/wiki/Cellulose" title="Cellulose">cellulose</a>  
  <a href="/wiki/Fibre" class="mw-redirect" title="Fibre">fibres</a>  
  " derived from "  
  <a href="/wiki/Wood" title="Wood">wood</a>  
  ", "  
  <a href="/wiki/Textile" title="Textile">rags</a>  
  ", "  
  <a href="/wiki/Poaceae" title="Poaceae">grasses</a>  
  ", or other vegetable sources in "  
  <a href="/wiki/Water" title="Water">water</a>  
  ", draining the water through a fine mesh leaving the fibre evenly  
  distributed on the surface, followed by pressing and drying. Although  
  paper was originally made in single sheets by hand, almost all is now made  
  on large machines—some making reels 10 metres wide, running at 2,000  
  metres per minute and up to 600,000 tonnes a year. It is a versatile  
  material with many uses, including "  
  <a href="/wiki/Printing" title="Printing">printing</a>  
  ", painting, graphics, signage, design, packaging, decorating, "  
  <a href="/wiki/Writing" title="Writing">writing</a>  
  ", and "  
  ...  
  
<div data-bbox="111 785 878 805" style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;">jet-targetContainerdiv#mw-content-text.mw-body-contentdiv.mw-content-ltr.mw-parser-outputpa


```


Styles	Computed	Layout	Event Listeners	DOM Breakpoints	Properties	Accessibility
--------	----------	--------	-----------------	-----------------	------------	---------------

Filter:hov .cls+

ConsoleWhat's NewX

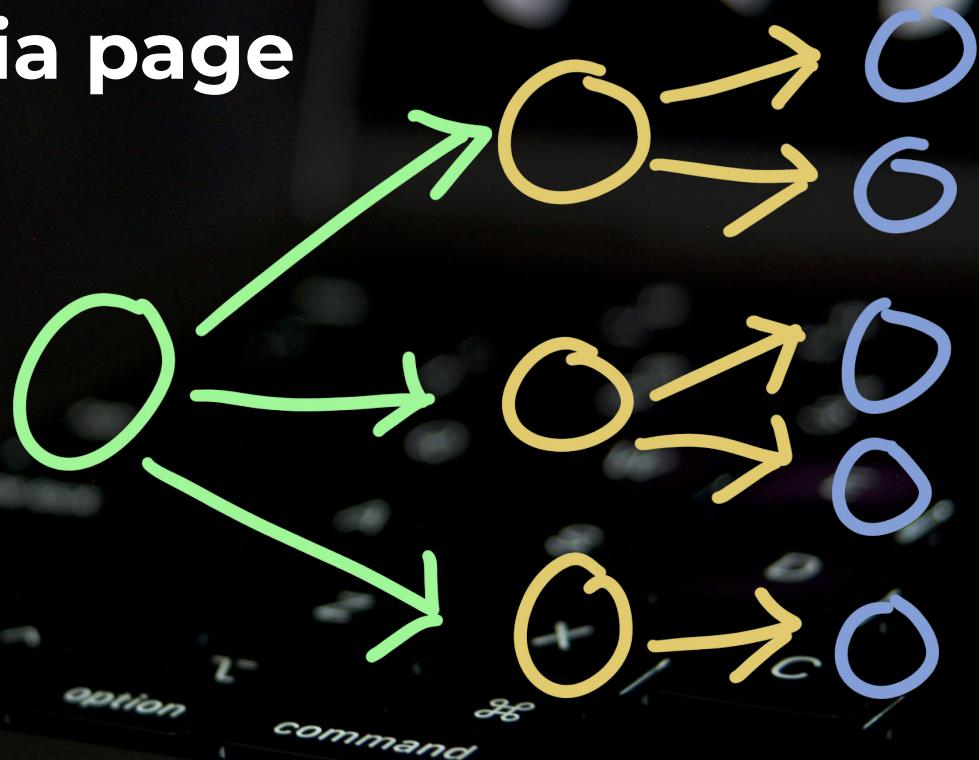
Highlights from the Chrome 120 updateX

> Scrape websites with BS4

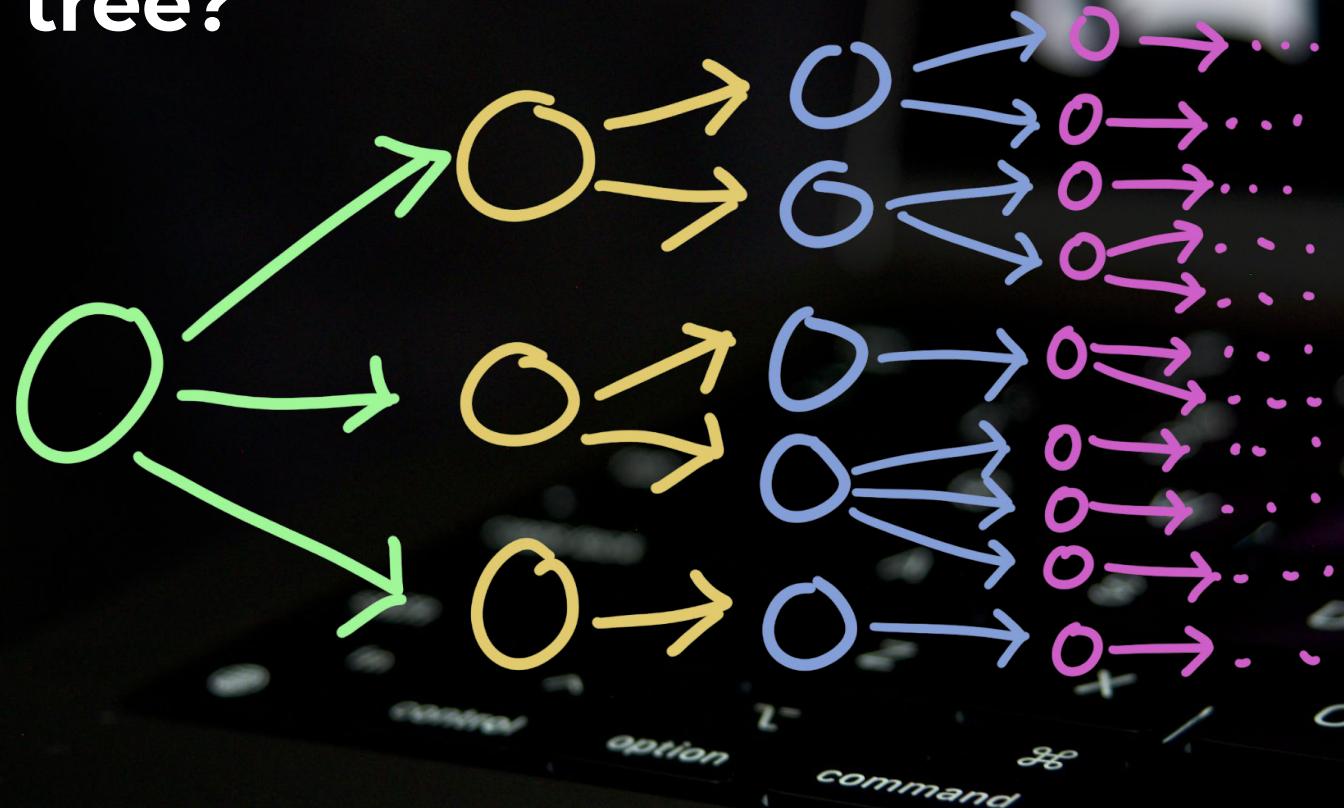
```
# Creates BeautifulSoup object from wikipedia link
def make_soup(self, wiki_topic):
    html = urlopen(f"https://en.wikipedia.org/wiki/{wiki_topic}")
    soup = BeautifulSoup(html, "html.parser")
    return soup

# Finds topics within Wikipedia page
def find_new_topics(self, soup):
    topics = []
    for link in soup.select('p a[href]'):
        href = link['href']
        if href.startswith('/wiki/'):
            topics.append(href[6:])
    return topics
```

> Now we can find links
that lead out from the
current Wikipedia page

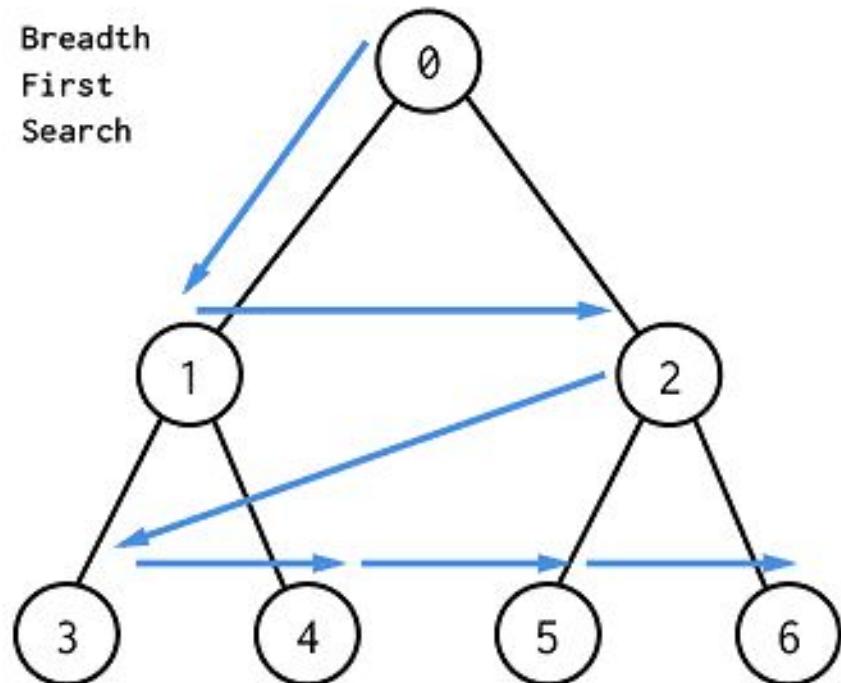


> How should we search this tree?

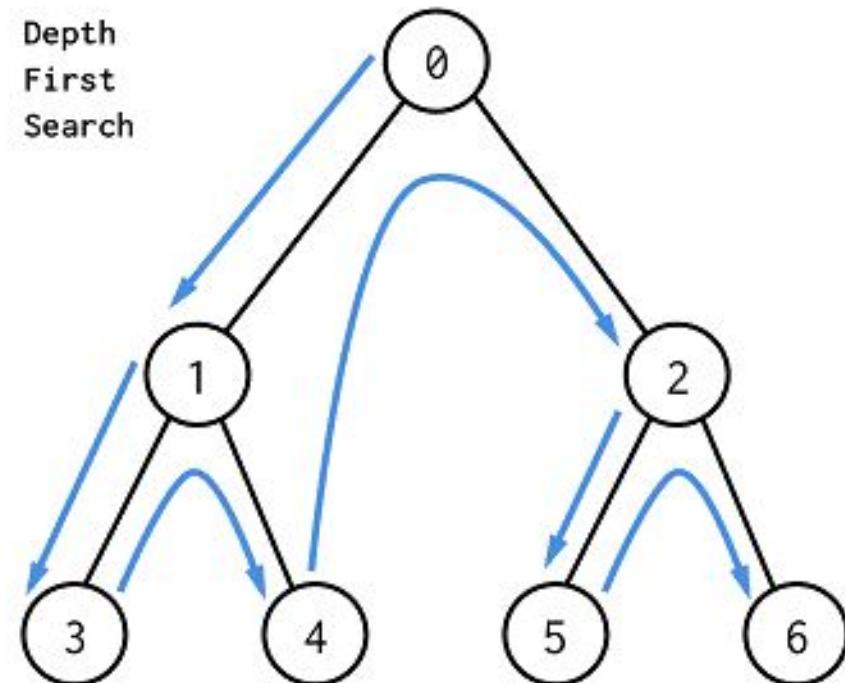


> BFS vs DFS

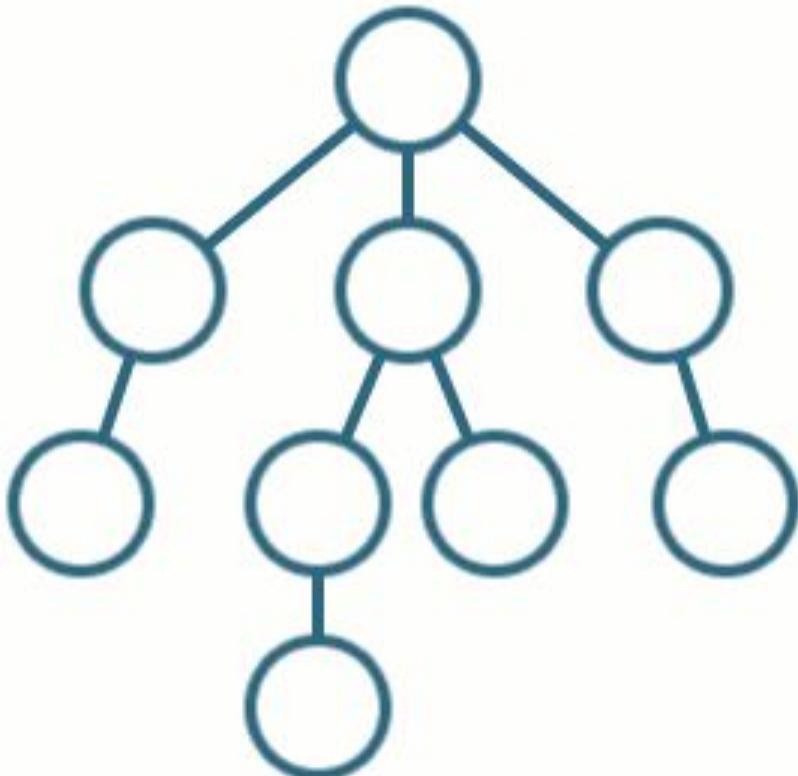
Breadth
First
Search



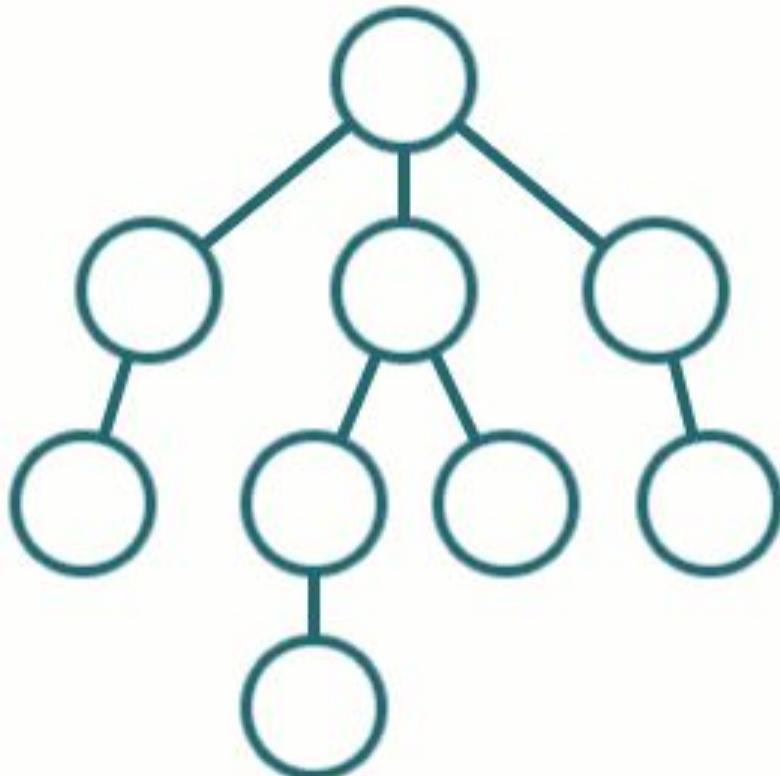
Depth
First
Search



DFS



BFS

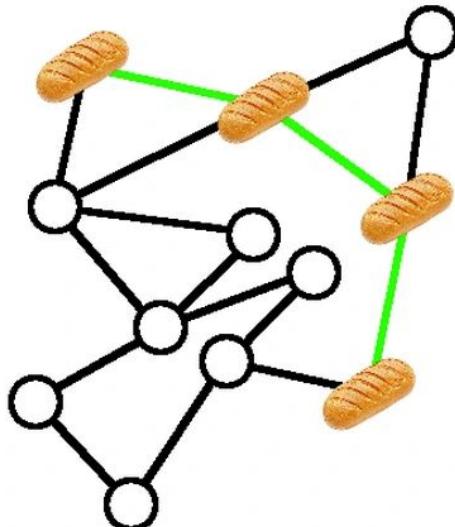


> BFS vs DFS

Bread



Bread-th First Search



Johnny Depp



Johnny Depp-th First Search

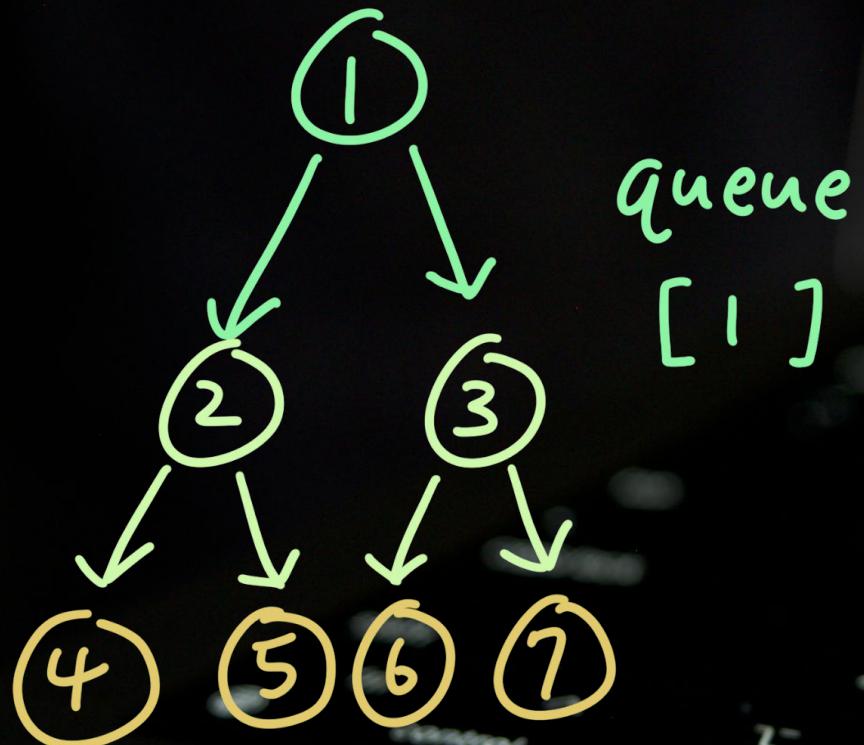


> How do we code this?

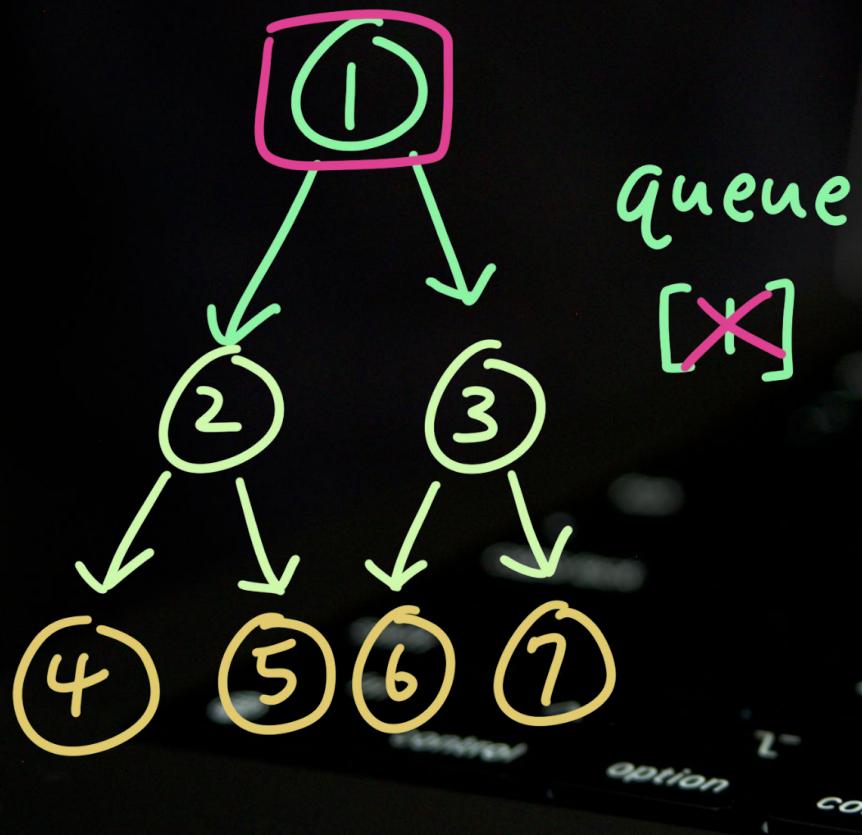
By using a queue!



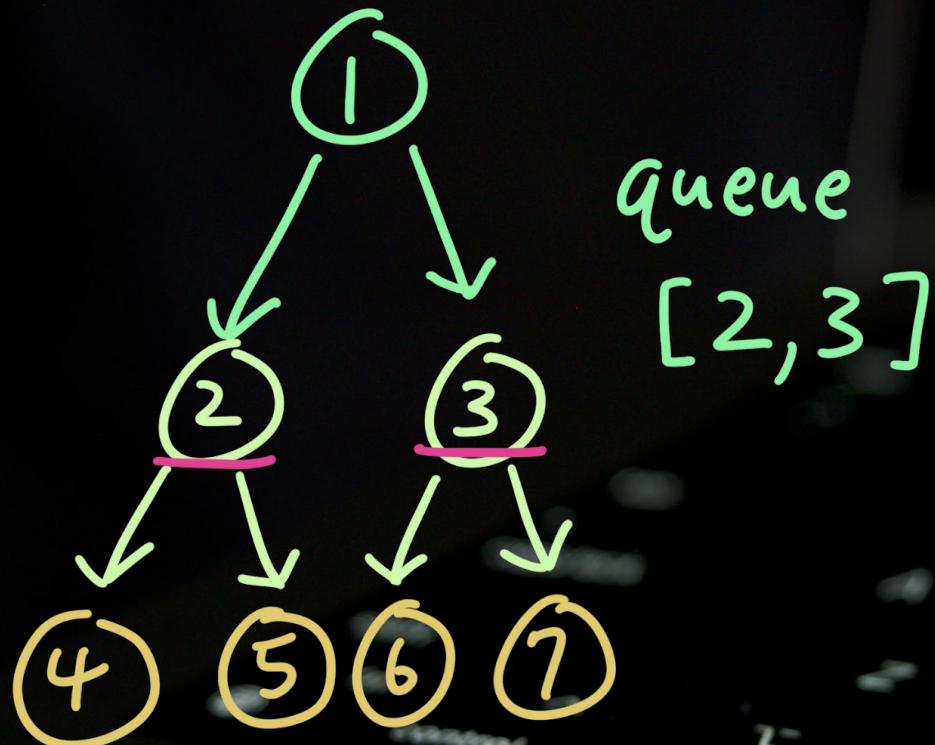
> How do we code this?



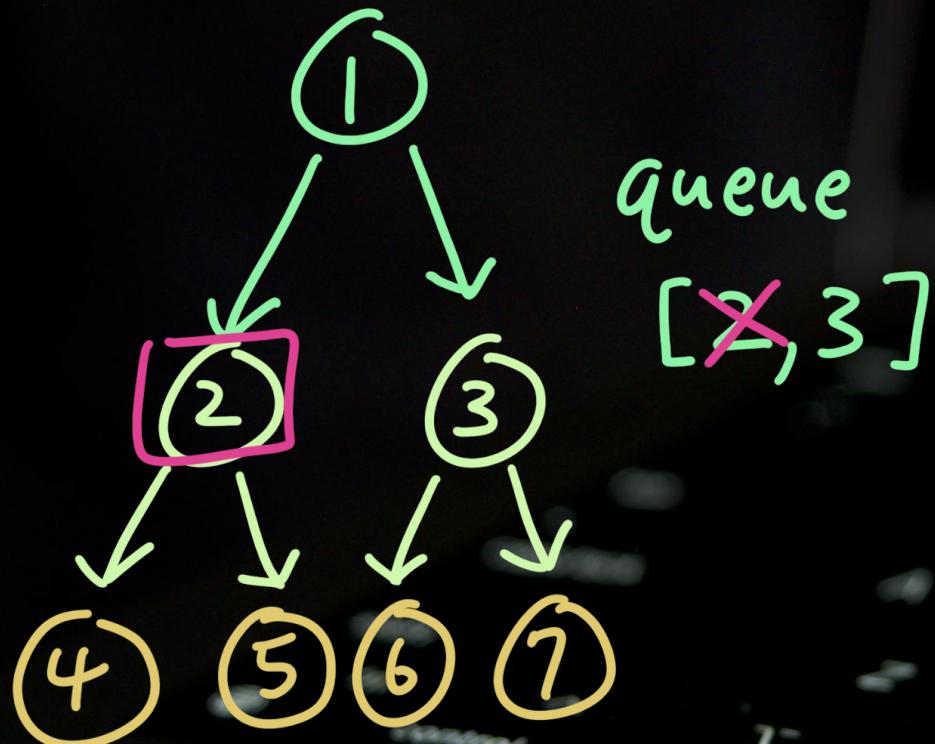
> How do we code this?



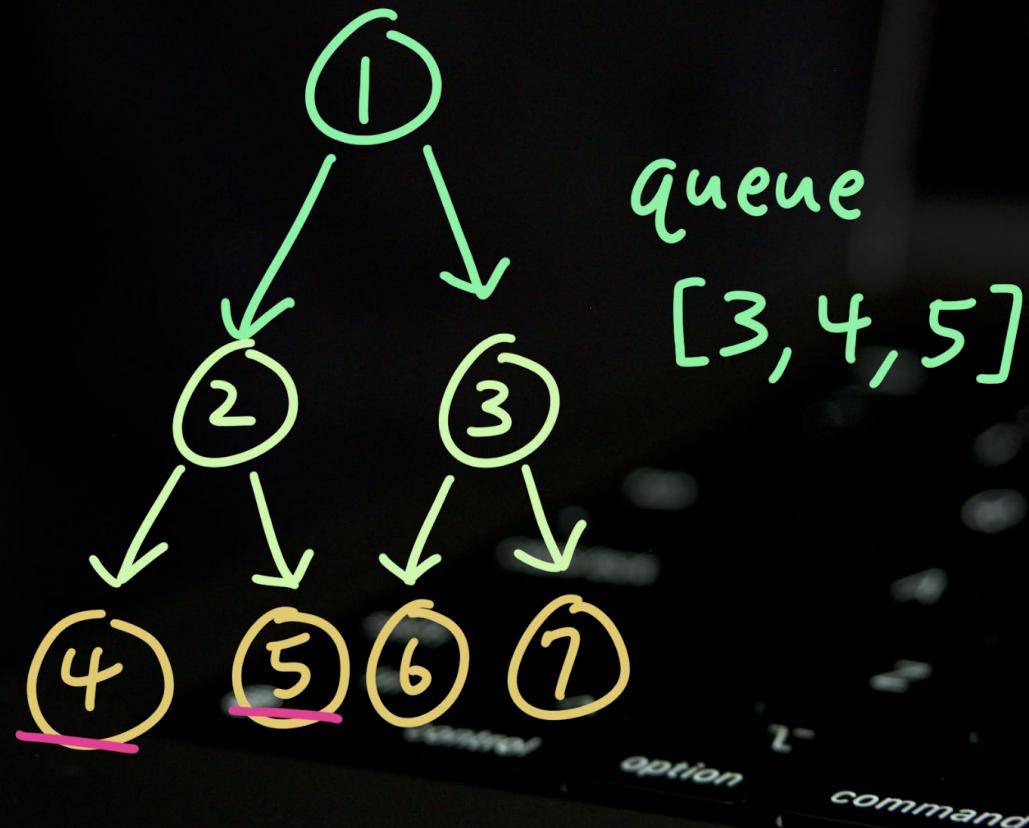
> How do we code this?



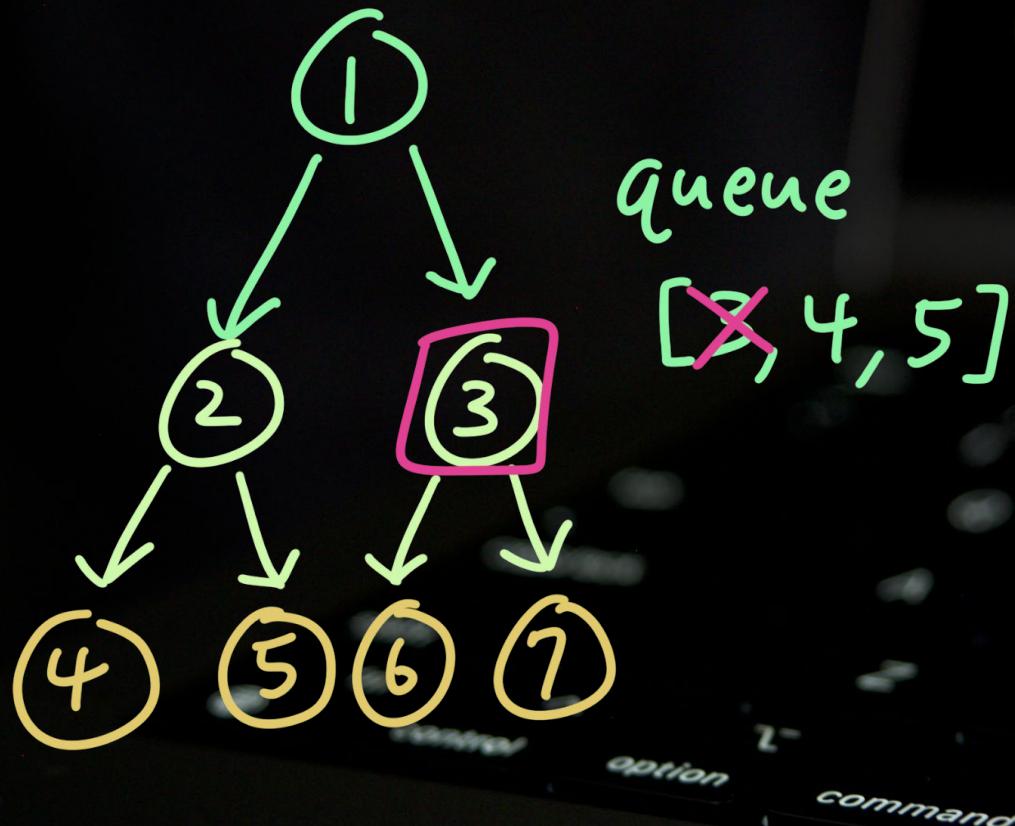
> How do we code this?



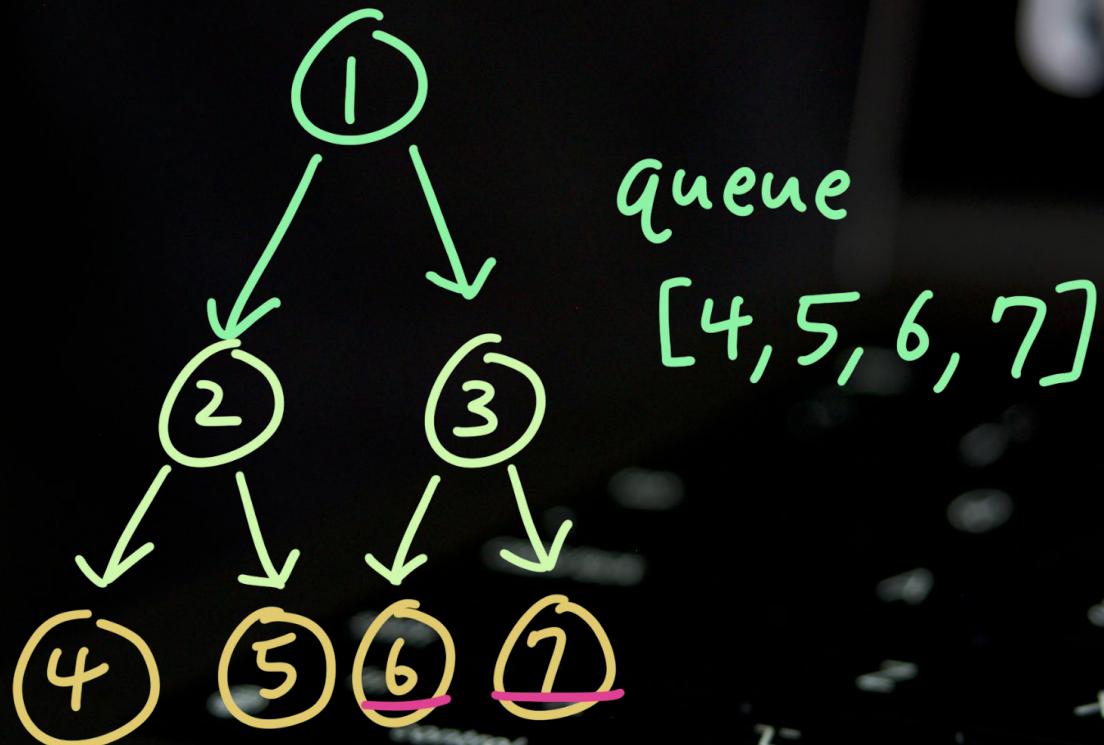
> How do we code this?



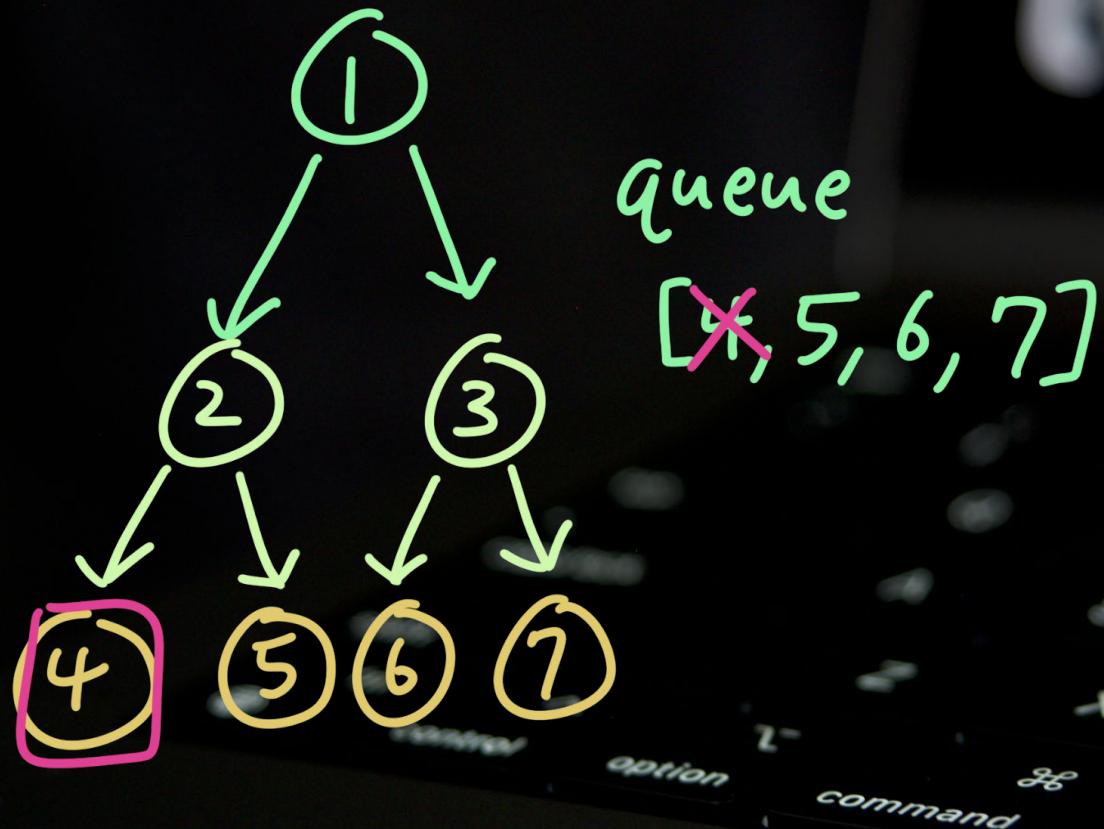
> How do we code this?



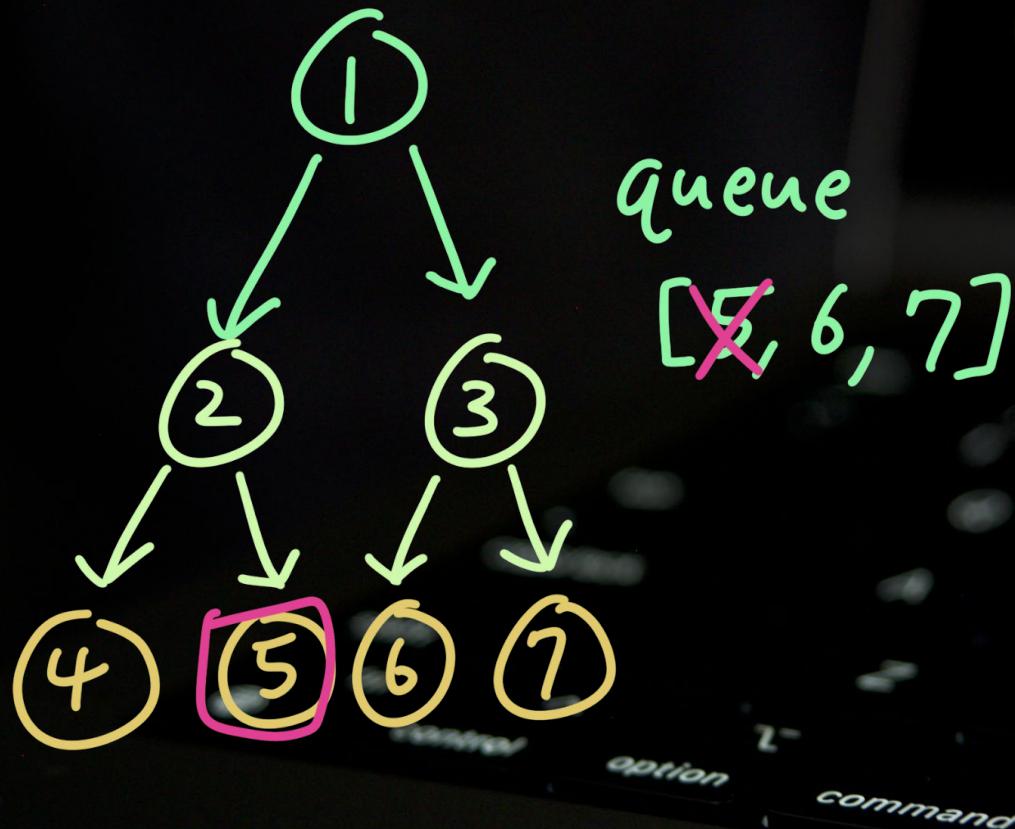
> How do we code this?



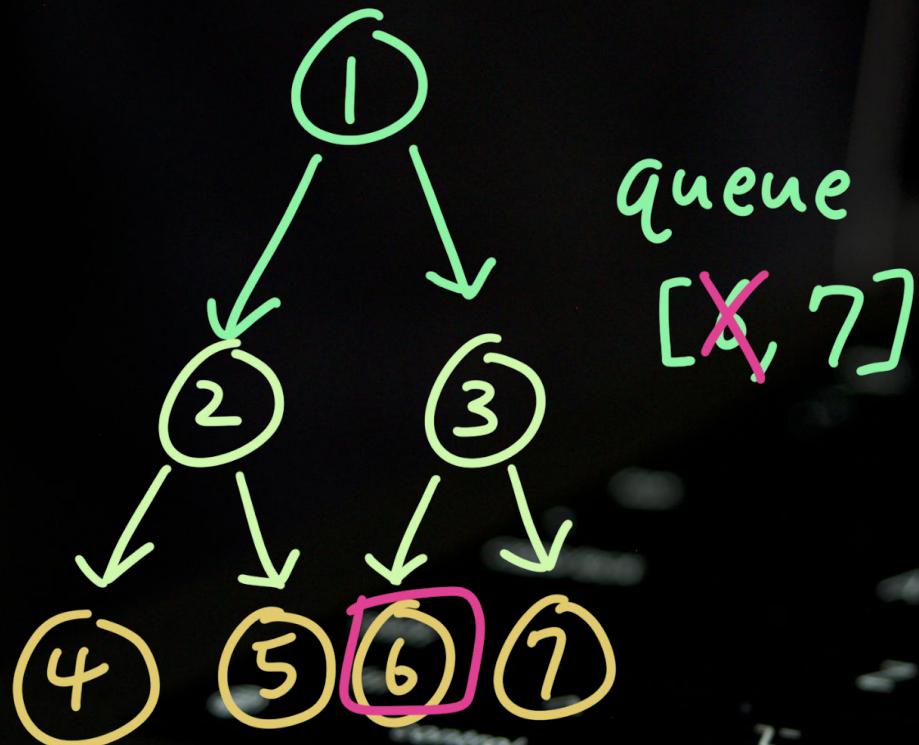
> How do we code this?



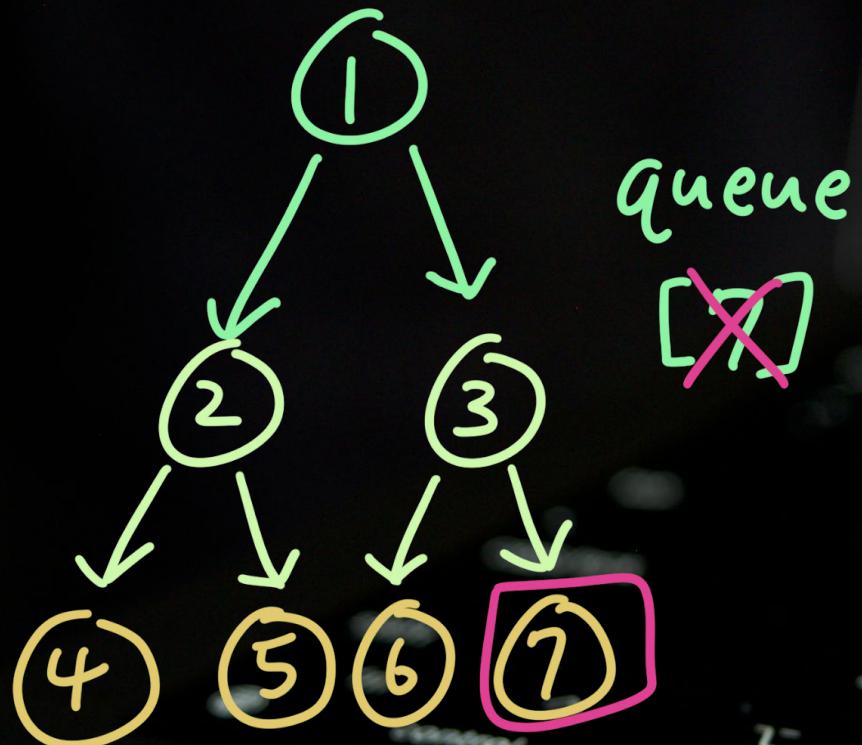
> How do we code this?



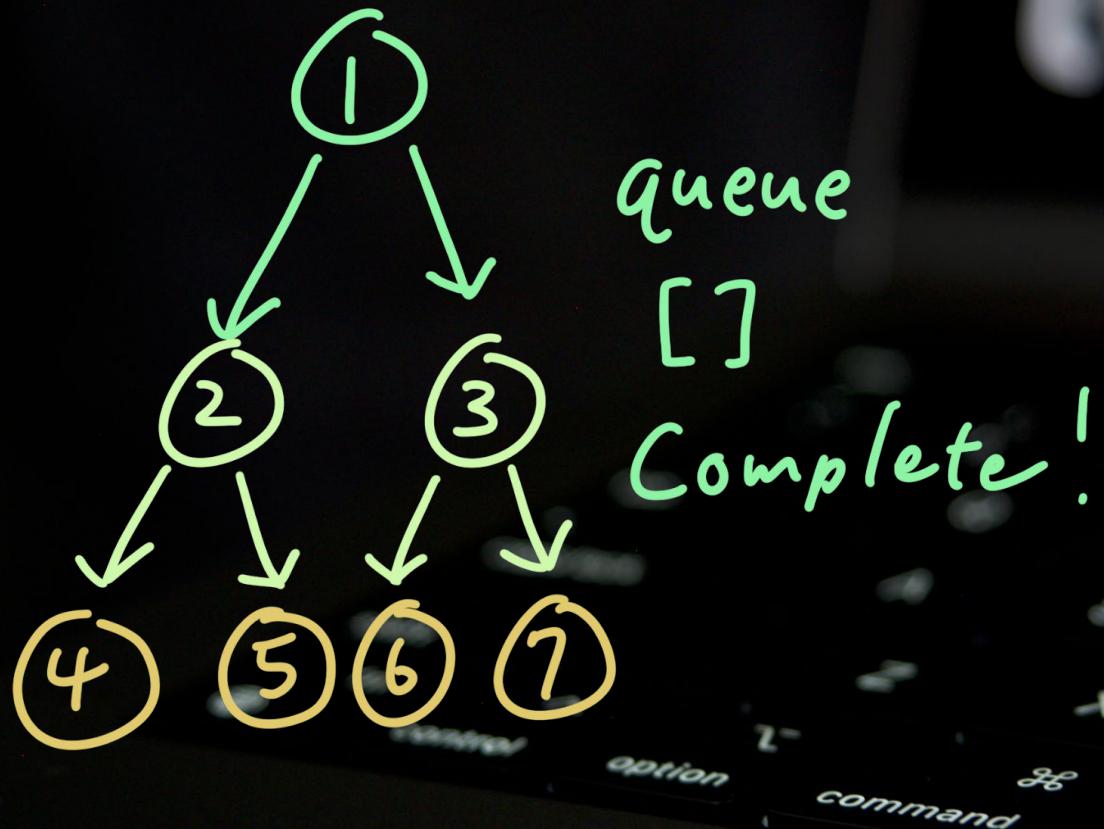
> How do we code this?



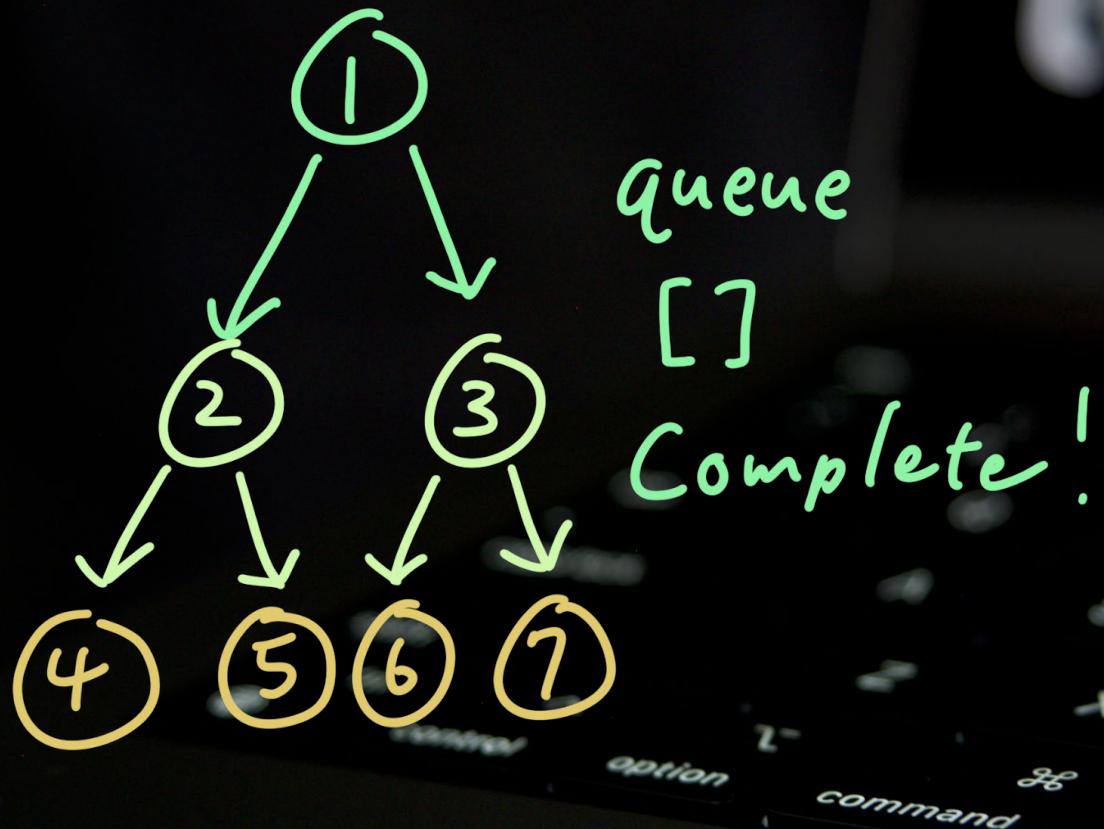
> How do we code this?



> How do we code this?



> How do we code this?



> Let's take a look at the code

```
class WikiGame:  
    def __init__(self, spread_constant=10):  
        self.visited_topics = []  
        self.path = []  
        # Determines how many links it grabs from each wikipedia page  
        self.spread_constant = spread_constant
```

> Let's take a look at the code

```
def start_game(self):
    # Asks user for start and end links
    start_link = input("What is the start link?:")
    self.start_topic = re.findall("\/wiki\/( [^#]*", start_link)[0]
    goal_link = input("What is the goal link?:")
    self.goal_topic = re.findall("\/wiki\/( [^#]*", goal_link)[0]

    print("Search Started")
    start_time = time.time()
    # Begins Breadth First Search
    discovered_path = self.bfs_search()

    print("\nPath Found:")
    for topic in discovered_path:
        print(topic)
    print(f"\nTime taken: {time.time()-start_time:.2f} seconds")
```

> Let's take a look at the code

```
def bfs_search(self):
    # Begins queue
    queue = [(-1, self.start_topic)]

    while queue:
        # Takes first topic in queue
        parent_index, parent_topic = queue.pop(0)

        if parent_topic not in self.visited_topics:
            #
                # Add parent_topic to the list of visited topics
                # Hint: self.visited_topics.append()

            #
                self.path.append(parent_index)

                # Gets topics from the Wikipedia page
                soup = self.make_soup(parent_topic)
                topics_found = self.find_new_topics(soup)

                print(f"{parent_index} From {self.visited_topics[parent_index]}, searching through: {parent_index}
```

> Let's take a look at the code

> Let's take a look at the code

```
[ ]: game = WikiGame()  
game.start_game()
```

> Socials

Website: hhscs.club

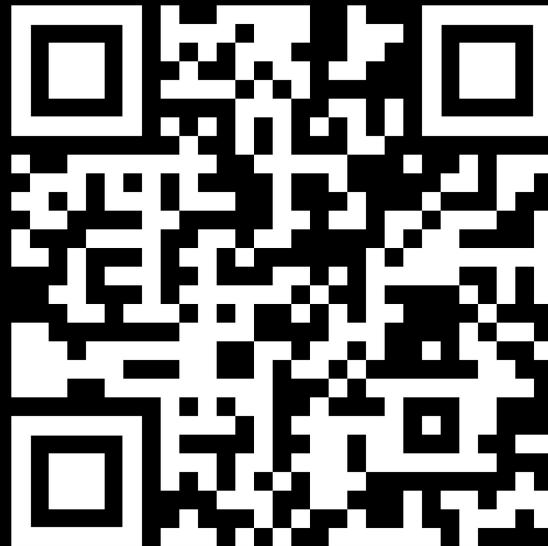
Email List:



Insta:

@hhscscomputerscience

Discord:





Next Meeting:
Tuesday (1/30) Lunch