

Press your forehead everytime you copy code from stackoverflow



First year computer science kids



```
6 bool is_prime(int x)
7     return false;
8 }
```

test no. 99989: passed
test no. 99990: passed
test no. 99991: failed
test no. 99992: passed
test no. 99993: passed
test no. 99994: passed
test no. 99995: passed
test no. 99996: passed
test no. 99997: passed
test no. 99998: passed
test no. 99999: passed
95.121% tests passed

1 in 8 Americans
have worked at McDonald's

1 in 10 Americans
think HTML is
a sexually transmitted disease.

40 people have
only done 1 of the
3 required surveys:
Check brightspace
announcement for
links

When you close more than 100
tabs after finding solution of a bug



```
print("X X X X X X X X X X")
print("I I I I I I I I I I")
print("O O O O O O O O O O")
print("O O O O O O O O O O")
print("O O O O O O O O O O")
print("I I I I I I I I I I")
print("X X X X X X X X X X")

print("Your turn! 1.")
player = input()

if player == "e4":
    print("X X X X X X X X X X")
    print("I I I I I I I I I I")
    print("O O O O O O O O O O")
    print("O O O O O O O O O O")
    print("O O O O O O O O O O")
    print("I I I I I I I I I I")
    print("X X X X X X X X X X")
elif player == "d4":
    print("X X X X X X X X X X")
    print("I I I I I I I I I I")
    print("O O O O O O O O O O")
    print("O O O O O O O O O O")
    print("O O O O O O O O O O")
    print("I I I I I I I I I I")
    print("X X X X X X X X X X")
elif player == "e3":
    print("X X X X X X X X X X")
    print("I I I I I I I I I I")
    print("O O O O O O O O O O")
    print("O O O O O O O O O O")
    print("O O O O O O O O O O")
    print("I I I I I I I I I I")
    print("X X X X X X X X X X")
```

if a mf pull a on me and
say

4. Median of Two Sorted Arrays

Hard 22.2K 2.6K

Amazon Bloomberg Google

Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return the median of the two sorted arrays.

The overall run time complexity should be $O(\log(m+n))$.

tell my mama i love her

Graphs: CS251 (Spring 2025)

CC/SCC, Tarjans

Andres Bejarano
(Instructor)

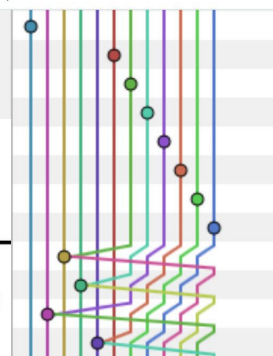
Ethan Dickey
(Guest Lecturer)



BANK\$
@Polycarboss

Learning JavaScript is like looking both ways
before you cross the street, and then getting hit by
an airplane.

I fcked up Git so bad it turned into Guitar Hero

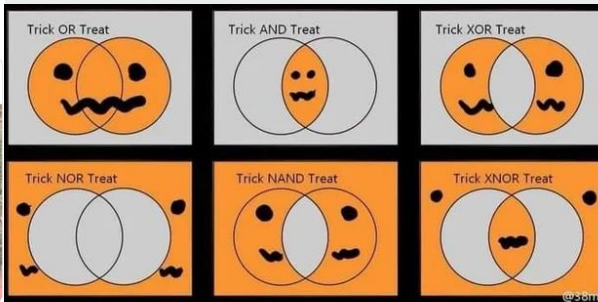


Supabase
@supabase

British devs after typing "color" in CSS



make sure to clean your keyboard every month



GIT INIT

GIT, INNIT?

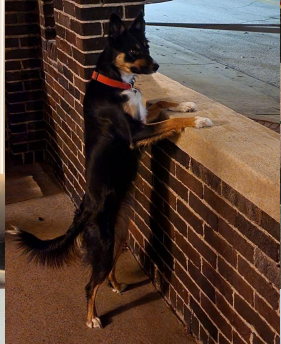
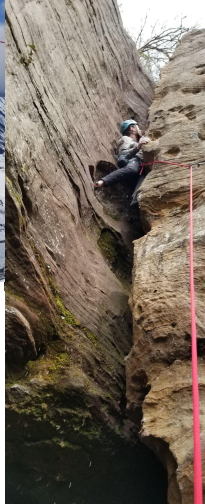
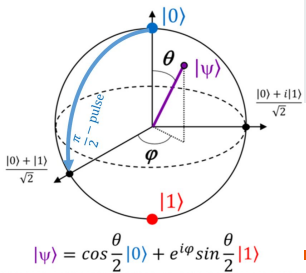
CC/SCC Outline



Today!

- CCs
 - Counting and finding the CCs
- SCCs
 - Counting and finding the SCCs: Trajan's Algorithm
- ChatGPT o.O
- Course policy updates

Who's this guy?



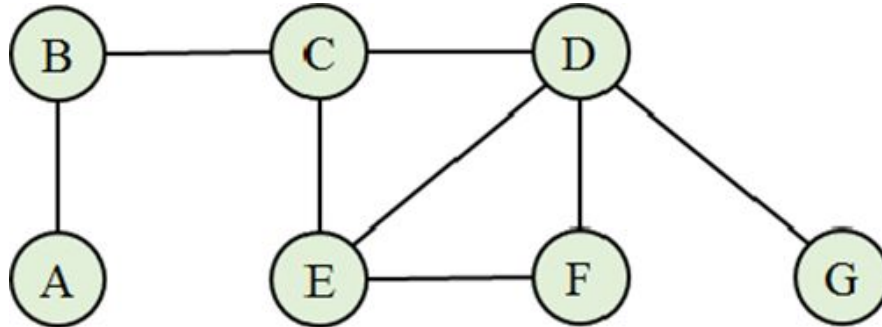
- 4th year PhD student
 - Researching Quantum Game Theory!
- 7th semester as the Instructor of CS211 CP1
- Love to climb, hike 14ers (Colorado), read, ski, play board games, and yes, I love to code
- I have 2 dogs :)
- Oh, and I do research into CS Education + GenAI



Connected Component (CC)/ Strongly Connected Component (SCC)

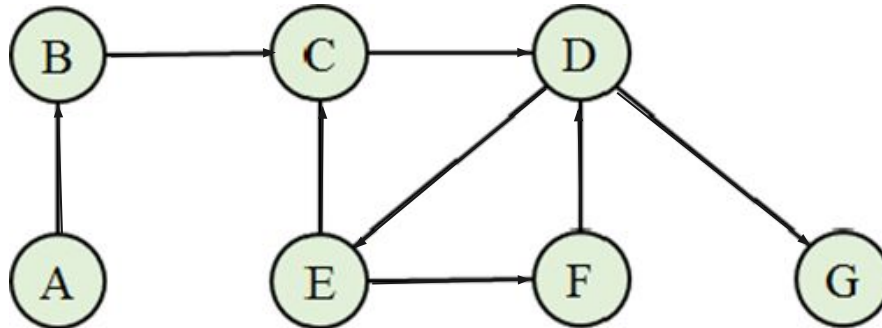
Connected Components

- In an undirected graph, a cycle is “a non-empty trail in which only the first and last vertices are equal”
- For undirected, slightly different term: Connected Component
 - More formally, a set of vertices are a *connected component* in a graph if any vertex in the set can be reached by any other vertex in the set by traversing edges.
- What are the connected components in this graph?
 - All of the nodes are in the same connected component! (a, b, c, d, e, f, g)
- What if we remove the edge (b, c)?
 - (a, b) and (c, d, e, f, g) are the 2 connected components
- What if we then remove the edge (e, d)?
 - Nothing changes because we can still reach all of (c, d, e, f, g) via edges

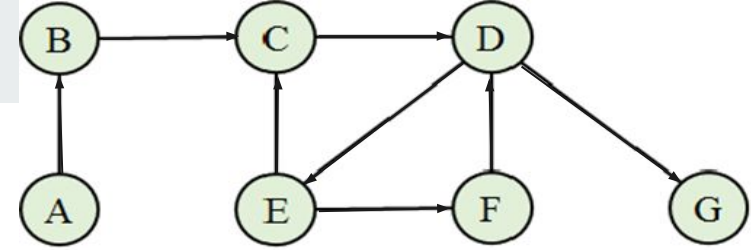


Strongly Connected Components

- AKA identifying cycles in a directed graph
- More formally, a set of vertices are a *strongly* connected component in a *directed* graph if any vertex in the set can be reached by any other vertex in the set by traversing edges.
- What are the strongly connected components in this graph?
 - (a), (b), (c, d, e, f), (g) <- why is G separate?
- What if we remove the edge (b, c)?
 - (a), (b), (c, d, e, f), (g) <- nothing changes!
- What if we then remove the edge (d, e)?
 - (a), (b), (c), (d), (e), (f), (g)



Tarjan's Algorithm



- Finding SCCs in $O(V+E)$ using *only one DFS* (Kosaraju's uses two DFSs, we won't be teaching it here)
- Basics are as follows:
 - We perform a DFS on every node if it hasn't already been visited (as all DFS-based algorithms start)
 - We keep a record of nodes that we've seen in our current DFS in a stack. Each SCC will be found based on when we visited certain nodes ("discovery index") and where we detected a cycle
- Implementation:
 - Each vertex v has:
 - A "discovered" index which numbers nodes in order of discovery
 - A "lowest link" value which represents the lowest-indexed node on the **stack** reachable from v 's DFS subgraph, including itself
 - We deal with the stack in the following way:
 - When we enter an undiscovered node, we push it onto the stack
 - When we leave a node in DFS, we either
 1. Keep it on the stack, if V or something that V can reach (later in the stack) can reach something that came *before* V (and therefore *earlier* on the stack than V)
 2. Remove it from the stack, if nothing that came after V (i.e. V can reach it via DFS) can reach something earlier on the stack (implies a cycle since "something earlier" can reach V). We also remove everything that V can reach from the stack (which would be later in the stack)
 - Therefore, V must be left on the stack if $v.\text{lowestLink} < v.\text{index}$, and removed if $v.\text{lowestLink} == v.\text{index}$

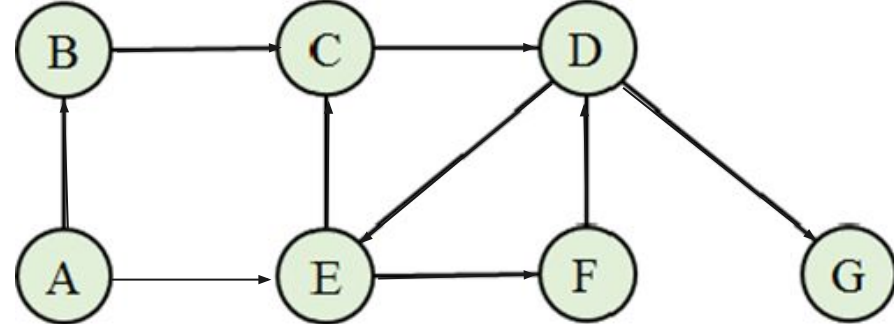
Pseudocode: Tarjan's Algorithm

```
function strongconnect(v):  
    // Set the depth index for v to the smallest unused index  
    v.index := index  
    v.lowlink := index  
    index := index + 1  
    S.push(v)  
    v.onStack := true  
  
    // Consider successors of v  
    for each (v, w) in E do  
        if w.index is undefined then  
            // Successor w has not yet been visited; recurse on it  
            strongconnect(w)  
            v.lowlink := min(v.lowlink, w.lowlink)  
        else if w.onStack then  
            // Successor w is in stack S and hence in the current SCC  
            // If w is not on stack, then (v, w) is an edge pointing to an SCC already found and must be ignored  
            // Note: The next line may look odd - but is correct.  
            // It says w.index not w.lowlink; that is deliberate and from the original paper  
            v.lowlink := min(v.lowlink, w.index)  
        end if  
    end for  
  
    // If v is a root node, pop the stack and generate an SCC  
    if v.lowlink = v.index then  
        start a new strongly connected component  
        Repeat  
            w := S.pop()  
            w.onStack := false  
            add w to current strongly connected component  
        while w ≠ v  
        output the current strongly connected component  
    end if  
end function
```

Calling Code:

```
input: graph G = (V, E)  
output: set of strongly connected  
        components (sets of vertices)  
  
index := 0  
S := empty stack  
for each v in V do  
    if v.index is undefined then  
        strongconnect(v)  
    end if  
end for
```

DFS	Stack	node: (index, lowlink)
Explore a	(a)	a: (0, 0)
Explore b**	(a, b)	b: (1, 1)
Explore c	(a, b, c)	c: (2, 2)
Explore d	(a, b, c, d)	d: (3, 3)
Explore g**	(a, b, c, d, g)	g: (4, 4)
Done @g	(a, b, c, d)	SCC++: {g}
Explore e	(a, b, c, d, e)	e: (5, 5)
Explore f**	(a, b, c, d, e, f)	f: (6, 6)
Find d		f: (6, 3)
Done @f	(a, b, c, d, e, f)	e: (5, 3)
Find c		e: (5, 2)
Done @e	(a, b, c, d, e, f)	d: (3, 2)
Done @d	(a, b, c, d, e, f)	

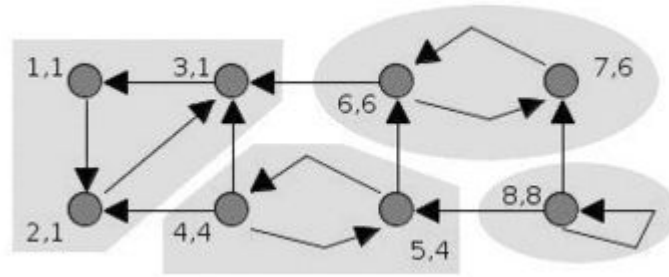


DFS	Stack	node: (index, lowlink)
Done @c	(a, b)	SCC++: {f, e, d, c}
Done @b	(a)	SCC++: {b}
Find e		(Ignore, not on stack)
Done @a	()	SCC++: {a}

Leave f on the stack because it points to d

**Node visit preference is arbitrary (and a bit contrived) here for academic purposes

Another animation



Graph algorithm resources

Play with algorithms here: <https://visualgo.net/en>

GPT of the Chat Variety

GenAI Uses

What is it good for?

What about in education?

- Conceptual questions
- Personal tutor
- Generating practice questions and test cases
 - <https://chat.openai.com/share/c7b2acfc-9bf1-4202-95e1-544eab7fcc0c>
- Re/writing/drafting/narratives/motivating examples

Educational no-nos?

- Asking homework questions or project questions without understanding the topics
- Believing everything it says (confidence in mistakes)
 - <https://chat.openai.com/share/20de44cb-7c29-47a1-88ce-db77e923c312>
 - If you can't figure out why this is wrong, see the notes on this slide

Personal tutor?



We would strongly encourage you to use the tools with phrases as follows:

- Help me with X, don't give me all the answers but help lead me to it
- Provide suggestions for how to start without solving my problem
- Find my bug but don't tell me exactly where it is, help me discover it for myself
- Only give me a partial solution to my problem, don't give it all away but guide me to the answer
- Here's my problem, help me find the issue by asking me guiding questions only

How can we use GenAI well?

Perspective: Interaction with the tools



Clear, concise, and specific

- 4-6 year old child
- Converse as you would with a human
 - “the part about X wasn’t very clear, could you explain it differently, please?”
 - *Language model*

Expect imperfection, frequently

- Ask for 10 things, expect 2-4 that match requirements

Prompt ~Engineering~ (for educational purposes)



1. Context
 - a. Expert(s) to act as (persona prompt pattern)
 - b. Setting
 - i. “Act as a political science professor teaching a summer school for gifted high school students”
 - ii. “Act as a political science professor teaching a summer school for academically advanced high school students”
 - c. Audience base knowledge
 - d. Together: “act as a collegiate professor with a PhD in Computer Science for a sophomore level data structures and algorithms course. Students are familiar with java.”
 2. Specific, measurable request: “give me a set of 10 problems to practice implementing Huffman Coding.”
- 2.5 Goals (optional – if not simple or not short enough)

Interacting with the response



- Most importantly: it is a *chatbot*
- Communicate likes and dislikes
- Separate specific refinements into individual queries
 - Do not say “Make 2 sound more like a story, make 3 clearer in the prompt, make 4 harder.”
 - “Context bleeding”
- (sometimes) Remind it of the context
 - (if needed) Start a new chat, remind it of where you were

Your turn! [pair programming]

In your pairs, choose one of the three following options. I will call on groups to share what they did/what mistakes they saw.

1. Have it create a tarjan's example, run through the steps, spot its mistakes
 2. Create exam practice problems for Tarjan's algorithm with answers, critique either the question or the answer (or both)
 3. Create practice problems for writing pseudocode on a topic from this semester, using one of the algorithms you've been taught (I've been told you're bad at that)
- These are not meant to be hard nor big tasks, just have fun with it.
 - **Interaction**
 - Be clear, concise, and specific
 - Converse with it (communicate likes and dislikes)
 - Expect imperfection
 - (sometimes) start a new chat/remind context
 - **Prompt engineering techniques**
 - Context ("act as," setting, base knowledge)
 - Specific, measurable request
 - (sometimes) Goals

Can I use ChatGPT on my homework to get answers?

No^{*T}

*This week you will be required to use GenAI to generate a solution for you to critique on ONE (1) problem only.

^TWe encourage you to seek tutoring help and help with conceptual questions, using GenAI without copying from it.

Course Policy Updates (pay close attn)

Nothing changes

1. If you copy + paste from AI (or anywhere), it is STILL cheating.
 - a. Specifically, if you submit work which is not your own, it is plagiarism in accordance with the course syllabus.
2. We will still evaluate correctness, so be careful if you use the tools for help.
3. You will not have these tools on the exams, so prepare yourself accordingly!
4. You are still 100% responsible for your responses. You are NOT allowed to complain if AI gave you something wrong.

HW6-AI

- 2 problems, required to use GenAI only
- Participation in all 3 components (prelab + 3 surveys, this lecture, HW6-AI + 3 surveys) will get you full points for an extra homework assignment (basically for free)
 - Available today, have 1 week

What did we learn today?

- Graphs! Cycles! Directed vs undirected, connected and strongly connected components
- Difference between CC and SCC
 - SCC is for directed graphs, CC for undirected
- How do we find CCs?
 - Everything reachable from the current node – DFS or BFS, take your pick
- How do we find SCCs?
 - Tarjans – DFS once using a stack to keep track of what's in the current SCC
- How should you interact with chat-based GenAI?
 - Be clear, concise, and specific
 - Converse with it
 - Expect imperfection
 - (sometimes) remind it of the context or even start a new chat
- Prompt engineering techniques?
 - Context ("act as," setting, base knowledge)
 - Specific, measurable request
 - (sometimes) Goals
- What do Canadians call Canadian bacon in Canada?
 - In Canada, the term "Canadian Bacon" is not used. The cut of meat from the loin is referred to as "back bacon". In Southern Ontario, "Peameal Bacon", as it is referred to there, is a similar product; however it is not smoked but rather set in a brine.
- How tall is the tallest person in the world?
 - The tallest man living is Sultan Kösen (Turkey, b.10 December 1982) who measured 251 cm (8 ft 3 in) in Ankara, Turkey, on 08 February 2011. The part-time farmer was the first man over 8 ft (2.43 m) to be measured by Guinness World Records in over 20 years.

End of CC/SCC lecture