Section 2: Graph Traversal and Big-O

1. Big-$\mathcal{O}$-No

Put these functions in increasing order. That is, if f comes before g in the list, it must be the case that $f(n)$ is $\mathcal{O}(g(n))$. Additionally, if there are any pairs such that $f(n)$ is $\Theta(g(n))$, mark those pairs.

- $2^{\log(n)}$
- $2^{n \log n}$
- $\log(\log(n))$
- $2^{\sqrt{n}}$
- $3^{\sqrt{n}}$
- $\log(n)$
- $\log(n^2)$
- $\sqrt{n}$
- $(\log(n))^2$

Hint: A useful trick in these problems is to know that since $\log(\cdot)$ is an increasing function, if $f(n)$ is $O(g(n))$, then $\log(f(n))$ is $O(\log(g(n))$. But be careful! Since $\log(\cdot)$ makes functions much smaller it can obscure differences between functions. For example, even though n^3 is less than n^4, $\log(n^3)$ and $\log(n^4)$ are big-Θ of each other.

2. Write it Slicker: A Proof by Contradiction

Claim: For every directed graph G, if every node of G has out-degree at least 1, then G has a directed cycle.

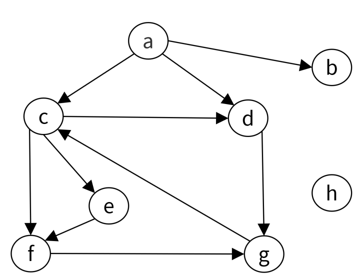
(a) Prove the claim using proof by contradiction.

(b) Rewrite the proof, using the proof by contradiction with extremality technique.

(c) There's another style yellow-flag in the version of this proof from part (a). We're proving an implication and our contradiction was the negation of one of the two things we supposed at the start. That usually means that proof by contrapositive would be clearer. Try writing this proof by contrapositive.

3. Mechanical: BFS and DFS

Consider the graph below.



(a) Run Breadth-First-Search on the graph below and number the layer for each node. Start with layer 0.

(b) Run Depth-First-Search on the graph below to classify the edges. Mark the start and end times for each vertex.

4. Graph Modeling

In this problem we're going to solve a classic riddle.

(a) First, you should solve the classic riddle yourself to get a feel for the problem.

> You are on the beach with a jug that holds exactly 5 gallons, a jug that holds exactly 3 gallons, and a large bucket. Your goal is to put **exactly** 4 gallons of water into the bucket. Unfortunately, the jugs don't have markings to tell how full they are (e.g., you can't just fill the larger jug $4/5$ full). What you can do are the following operations.
>
> - Completely fill any of your jugs.
> - Pour from one of your containers into another until the first container is empty or the second is full.
> - Pour out all the remaining water in a container.
>
> How do you get 4 gallons of water into the bucket?

(b) Now, let's write an algorithm to solve any instance of this puzzle. You are given a list of 10 jugs with (positive integer) capacities $c_1, ..., c_{10}$, ranging from 1 to C. Your goal is to determine whether it is possible to get exactly t gallons into a bucket with capacity that is at least t and at most B.

Hint: Think about how you can relate the possible "states" of the puzzle to the nodes of some graph. What would be a good way to define edges for this graph? How could you think of solutions to the puzzle in terms of the graph?

5. Judging Books by Their Covers

You have a large collection of books, and just got a new bookshelf. For aesthetic reasons, you're going to arrange your books by the color of their covers (not by author or subject). You wish to put only books of a single color on any given shelf. You have a list of pairs of books which you know to be the same color. This list might be only partial (it's possible that u, v, and w are all the same color, but your list might only have "u and v are the same color. w and v are the same color.", for example). You should assume that the "same color" relation is transitive.

Given your list, your job is to give an upper-bound on the number of shelves you need so that no shelf has more than one color of book. Describe an algorithm to give the best bound you can on the number of shelves needed.

You do not need a full proof of correctness, but you should describe the running time in terms of (whichever subset is appropriate): b, the number of books; p the number of pairs listed; s the number of shelves required (i.e., your final answer).