Graph Theory

Graph Theory

- So why do we need graphs?

Graphs are generally split into two categories: **directed** and **undirected** graphs. The difference between the two lies in the types of edges that are allowed in the graphs and the subsequent properties that it induces.

Directed Graphs

Directed graphs have what are called **directed edges**, where an edge between nodes A, B, denoted as (A, B), denotes a relationship **from A to B**. Note that this does not imply a relationship **from B to A**.

In an example where we have a directed graph that represents the relationships between instagram following relationships, where nodes are accounts/people and edges, denoted as (A, B), represents **A follows B**. While person **A** follows person **B**, there is no guarantee that person **B** follows person **A**. Therefore, it makes sense that a directed graph is used to represent "follow relationships" on Instagram.

- What is the greatest number of edges possible for a node in a directed graph given that there are V nodes in the graph?

Undirected Graphs:

Undirected graphs, on the other hand, have what we call **undirected edges**, where an edge between nodes A, B, denoted as (A, B), represents a relationship **between A and B.** Note that there is no notion of direction here.

- What could be a use case for undirected graphs?
- What is the greatest total number of edges possible in an undirected graph given that there are V nodes in the graph?

Some Definitions of commonly used words:

- In a directed graph, what are the **neighbors** of a particular node U?
- In an undirected graph, what are the **neighbors** of a particular node U?

Graph Search

Note: I assume familiarity with `BFS` and `DFS` here and try to provide some intuition on why BFS/DFS is used for their applications.

Breadth First Search

`BFS` is a graph search algorithm that, given a starting node, "radially expands outwards" from the starting node, thus visiting nodes in sorted order of their distance from the starting node.

- Knowing this property, does using BFS (with minimal modifications) **guarantee** that you'll find a shortest path from the starting node U to a node that's reachable from U given there is at least one shortest path?
- Can BFS find the number of total connected components in an undirected graph?
- How many times is each edge explored in full BFS in an undirected graph? What about when it's a directed graph?
- How many times is each node in full BFS visited?
- (True or False) In BFS on starting node U, node A is added to the queue before node B, that means `distance(U, A) < distance(U, B)`.

Depth First Search

`DFS` is another graph search algorithm that, given a starting node, will "explore a branch of the graph as deeply as possible before backtracking and exploring other branches."

- What is the time complexity of `Full_DFS`?
- (True or False) Using the implementation of `DFS` with clocks, where we mark each node's starting time and finish time, given a directed graph, if node V is reachable from node U and U is not reachable from V, is the following statement true: `finish[u] > finish[v]`
- Using the implementation of `Full_DFS` with clocks, when is the following statement true? Given two nodes U and V and we find U first, `start[u] < start[v] < finish[v] < finish[u]`
- Using the implementation of `Full_DFS` with clocks, when is the following statement true? Given two nodes U and V and we find U first, `start[u] < finish[u] < start[v] < finish[v]`

General Questions

- (True or False) Does every directed acyclic graph (DAG) has a topological ordering?
- If an algorithm 1 has a runtime of $\Theta(V^{1.5})$ and algorithm 2 has a runtime of $\Theta(E)$, make a case for when algorithm 1 will run faster than algorithm 2.
- Similarly, make a case for when algorithm 2 will run faster than algorithm 1

- A magician is doing a show. In the show, there are N hats. In each hat, there is one of K different colored birds. The magician then proceeds to show us that of the N hats, there are M pairs of hats that act as portals, where each bird is able to teleport to the other hat and back (teleport between each pair of hats). However, it turns out that the magician's trick fails if K different colored birds are able to teleport to the same hat. Help the magician in developing an algorithm that determines if it's possible that K different colored birds are able to teleport to the same hat, causing his trick to fail.