Summary Part

1. **In your own words, explain why you think it’s hard to visualise a large graph in a way that makes its structure and connections easy to understand.**

   Visualizing large graphs is challenging because of the sheer volume and complexity of the data they represent. As the number of nodes and edges increases, the graph becomes densely populated, leading to overlapping lines and clusters that are hard to distinguish. This clutter makes it difficult to identify individual connections, understand the overall structure, and discern meaningful patterns. Additionally, the layout algorithms that try to position nodes and edges in an aesthetically pleasing way often struggle to scale effectively, resulting in representations that are either too sparse or too congested.

2. **What is the maximum number of edges in an undirected graph? Hint: you only need one undirected edge between two nodes in an undirected graph…**

   In an undirected graph with n nodes.

$$\boldsymbol{\frac{n(n-1)}{2}}$$

3. **What is the purpose of the Queue in the BFS algorithm?**

   In the Breadth-First Search (BFS) algorithm, the queue is used to keep track of nodes that need to be explored. The BFS algorithm works by exploring each node level by level, starting from a given source node. The queue ensures that nodes are explored in the correct order:

   Nodes at the current level are processed first.

   Their unvisited neighbors are added to the queue.

   This process repeats, moving to the next level once all nodes at the current level have been explored.

   Using a queue helps maintain this level-order exploration, ensuring that nodes closer to the starting node are processed before those further away.

4. **Explain how keeping track of nodes already visited prevents an infinite loop in BFS.**

   In BFS, keeping track of nodes already visited is crucial to prevent the algorithm from entering an infinite loop. This is done by maintaining a record of all nodes that have been explored. Here's how it helps:

   Marking as Visited: When a node is first encountered, it is marked as visited.

   Avoiding Reprocessing: Before adding a node to the queue for exploration, the algorithm checks if it has already been visited.

   Stopping Redundant Work: If a node has been visited, it is skipped, ensuring that each node is processed only once.

   This method prevents the algorithm from revisiting the same nodes endlessly, thus avoiding infinite loops and ensuring that BFS terminates correctly after exploring all reachable nodes from the starting point.

Self-reflection part

What did you learn?

This week, I learned the mathematical properties of undirected graphs, and the operational mechanics of the Breadth-First Search (BFS) algorithm.

What went smoothly?

The logical structure of BFS and the mathematical clarity behind calculating maximum edges in undirected graphs were particularly straightforward.

What was difficult about the content this week?

Understanding how to efficiently layout and display large-scale graphs without losing clarity was complex.

How will you approach things differently next time?

Experiment more with different graph visualization tools and algorithms to better handle large graphs.

Do you have any feedback about the content for this week?

Good!