Basic Graph Algorithms

Notation: $[n] = \{1, 2, 3, \dots, n\}$

n to denote # of vertices
m to denote # of edges

$G = (V, E)$

Directed vs Undirected

Paths vs Cycles vs Walks:

$\hookrightarrow$ A sequence of edges.

$v_{i_1}, \dots, v_{i_j}, v_{i_1}$: "closed path" (all distinct except for first and last)

$v_{i_1}, v_{i_2}, \dots, v_{i_j}$ where $v_{i_\ell} \in V$ (all distinct)
and for all $\ell \in [1, j-1]$,
$(v_{i_\ell}, v_{i_{\ell+1}}) \in E$.

$\lim_{n \to \infty} \frac{m}{n^2}$

Recall: $s \rightsquigarrow t$ connectivity: Given a graph $G = (V, E)$
Check if there is a path from $\underbrace{s \text{ to } t}_{\text{given as input}}$ in G.

Qn: What did we learn in DSA course that helps us solve this?

Breadth First Search (BFS) $O(n+m)$

→ Start from s. (Layer $L_0 = \{s\}$)
→ Enqueue all the neighbours of s. Build a layer L_1 with all neighbours of s. Mark all these as visited.

→ $\forall j \geq 2$, Build a layer with those neighbours of vertices in L_{j-1} which have not been visited.

↳ Layer $L_j = \{\text{Neighbours of vertices in } L_{j-1}\} \setminus \bigcup_{k=0}^{j-1} L_k$

Mark vertices in L_j visited.

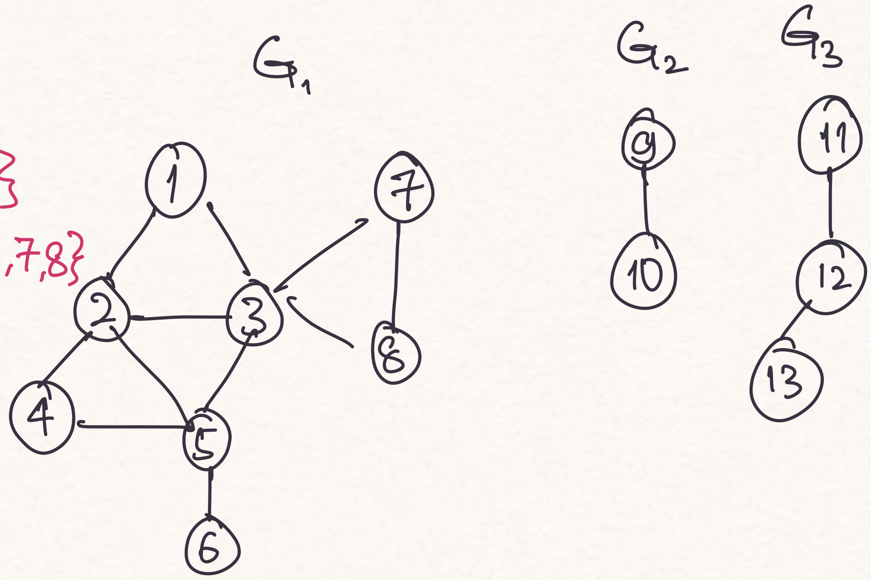
→ Repeat until all vertices are marked.* What if graph is disconnected.

$L_0 = \{1\}$

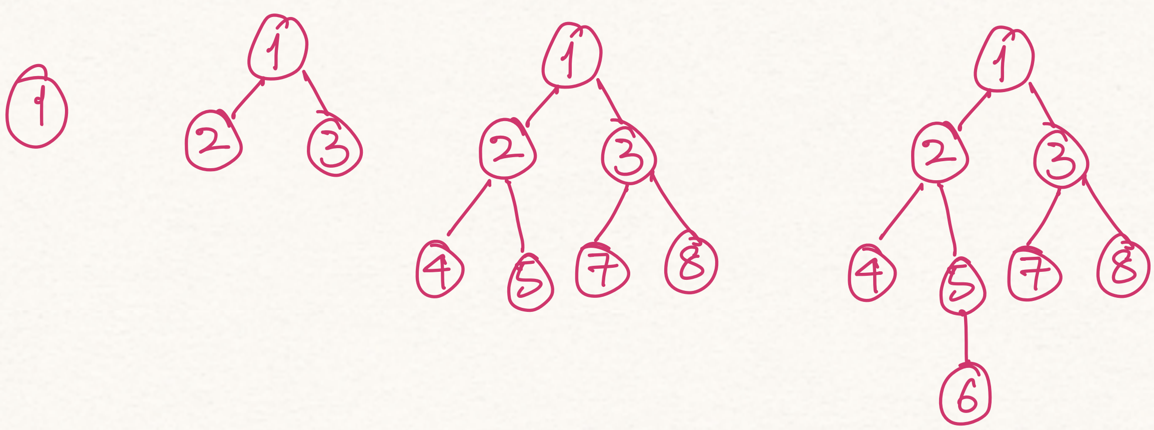
$L_1 = \{2, 3\}$

$L_2 = \{4, 5, 7, 8\}$

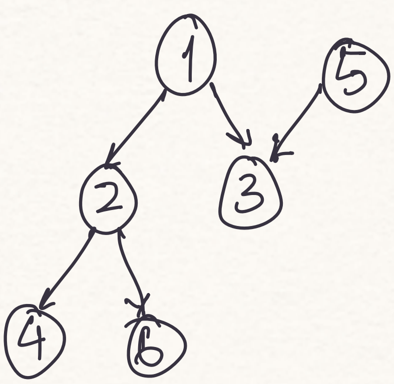
$L_3 = \{6\}$



$G = G_1 \cup G_2 \cup G_3$



BFS tree



Observation 1: BFS gives rise to a rooted tree structure.

Claim: $\forall j \geq 1$, layer L_j consists of all nodes that are at distance $= j$ away from s.

min   min length of a path from s to vertices in L_j

Proof: By induction:

Base case: L_1: All elements of L_1 are distinct from s and are connected by an edge.
$\Rightarrow$ all vert. of L_1 have a dist of 1 from s.

I.H: $\forall j \leq k$, we have that all elements of Layer L_j have a min dist of $= j$.

I. step: For the sake of contradiction, assume that $\exists$ a vertex t in L_{k+1} that has a min dist $< k+1$.

↳ We will now argue that t was considered and marked much before $k+1^{th}$ step

→ $s \longrightarrow t$ if $\exists$ a path of length $\ell < k+1$

then the algo would have added t into a layer$_j$, $j \leq \ell$.

→ $\nexists$ a vertex t s.t min dist of t from s $> k+1$.

$\Rightarrow$ $\forall$ vertices in Layer k, $\nexists$ a path of length $< k$ from s to those vertices.