CS2040S Tutorial 8

AY 24/25 Sem 2 — github/omgeta

Q1. (a) Since every node is visited at most once, it cannot possibly be visited from a different node from the parent node to form a cycle.

(b) Same algorithm can be used for both trees and graphs, but running on a non-root in a tree may cause nodes other than the subtree rooted at that non-root to be missed.

Q2. Solution:

1. BFS/DFS on any unvisited node, marking all nodes traversed as visited
2. When there are no more nodes, increment count of CC by 1 and repeat step 1

Q3. Solution:

1. BFS on n nodes, tracking edges encountered
2. Tree iff $n - 1$ edges for n nodes

Q4. (a.) 1. Directed graph with nodes as people and edges as spread
2. Special types of edges?

(b.) Bipartite graph with people and location edges.

(c.) ?

(d.) Undirected graph with nodes as students and edges as similarity. Any non-trivial components are cheaters.

(e.) Bipartite graph with children and preesent nodes, edges as desire. Greedy allocation with priority given to children with smaller desire.

Q5. (a.)

(b.) DFS, because resolving each letter first would give us more info for later letters.

(c.) Search finishes when there is only 1 edge.

Q6. Bipartite graph with nodes as students and edges between nodes if students are on the same card.
Consistency: check bipartitite with BFS/DFS in a way that $|G| > |B|$
Sufficiency: Insufficient if no possible permutations of flipping G, B for connected components leads to a consistent solution.
Assuming both consistent and sufficient, assign in a way that for each connected component, we prioritise $|G| > |B|$

Q7. ?