CSC 411 Quiz 2 — Solutions

1. If f and g are functions defined on the integers, what does it mean that a function $f = O(g)$?

   Solution. $f(n) = O(g(n))$ means there are constants $c > 0$ and n_0 such that for all $n \geq n_0$,

   $$0 \leq f(n) \leq c\, g(n).$$

   So, beyond some threshold, f is at most a constant-factor multiple of g.

2. When implemented with data structures instead of recursion, which data structure does DFS use? Which does BFS use? In your answer make it clear which method goes with which data structure.

   Solution.

   - **DFS** uses a **stack** (explicit stack in iterative DFS, or the call stack in recursive DFS).
   - **BFS** uses a **queue**.

   Reason: DFS explores most recently discovered vertices first (LIFO behavior), while BFS explores in discovery order by distance layers (FIFO behavior).

3. When searching for a node with a particular property, One of DFS and BFS may fail to terminate even if the node "should" be reachable. Which method has that limitation and what might cause it to happen?

   Solution. The limitation is with **DFS** (especially in infinite or very deep state spaces). DFS can follow one branch indefinitely and never return to explore the branch containing the target. So even when a target is reachable, DFS is not guaranteed to find it if it can get trapped in an infinite descent.

   By contrast, BFS explores level by level, so if the target is at finite distance, BFS reaches that distance in finite time.

4. In class we looked at pseudocode for BFS on an undirected graph, but as we discussed BFS also runs on directed graphs. To make it work, does anything need to be changed in the implementation of BFS? If so, write pseudocode for the modified algorithm.

   Solution. The core BFS structure does *not* change. The only conceptual change is that, for directed graphs, when processing vertex u we iterate only over *out-neighbors* of u. In adjacency-list representation this is already what 'Adj[u]' contains.