Lab09-exercises

Name: Minjie Shen

1. We have focused primarily on time complexity in this course, but when choosing data structures, space complexity is often as important of a constraint. Given an adjacency matrix, what is the 'space complexity' in Big-O. That is, given n nodes, how much space (i.e. memory) would I need to represent all of the relationships given. Explain your response.

For adjacency matrix, the space complexity if O(N * N) which is O(N^2). Since we have N nodes, and each node has a relationship with all the other nodes, which could be connected or not connected, represented as 1 or 0. So this matrix will be N rows and N columns, which is N^2 space. So the overall space complexity for adjacency matrix is O(N^2).

2. Will it ever make sense for ROWS != COLUMNS in an adjacency matrix? That is, if we want to be able to model relationships between every node in a graph, must rows always equal the number of columns in an adjacency matrix? Explain why or why not.

Yes, rows much always equals to columns in an adjacency matrix. An adjacency matrix shows how each node in a graph is connected to every other node in the graph. So, each row in the adjacency matrix represents "from node x", and each column represents "to node y". So the number of columns necessarily equals the number of rows.

3. Can you run topological sort on a graph that is undirected?

Yes, we can run topological sort on an undirected graph. However, it is ambiguous to run topological sort on an undirected graph. In an undirected graph, edges have no start and end point, and nodes either are mutually adjacent or mutually not adjacent. It is hard to decide a start point of the graph and the end point of the graph. At the same time, since the graph is undirected, node A can directs to node B and node B can directs to node A. Running topological sort on this undirected graph will create cycles, and the topological ordering will have an infinite loop, thus the ordering does not have any meaning.

4. Can you run topological sort on a directed graph that has cycle?

Yes, we can run topological sort on a cycled graph, but the sort will have meaningless. Since the graph has cycle, it is hard to decide the ordering. We can say any node in the cycle as a first node. However, no matter which point in the cycle you arrive at first, there is no possible way to satisfy the topological sort. In this case, the topological sort will be meaningless.

5. For question number 4, how would you know if you have a cycle in a graph? What algorithm or strategy could you use to detect the cycles? Hint we have already learned about this traversal.

We can use a depth first search to find a cycle in the graph. We put the first node in the stack, and marked this node as visited. Then we find all the vertices which are not visited and are adjacent to the current node. Recursively call the function for those vertices, If the recursive function returns true, return true. If the adjacent vertices are already marked in the recursion stack then we find a cycle, and we return true.