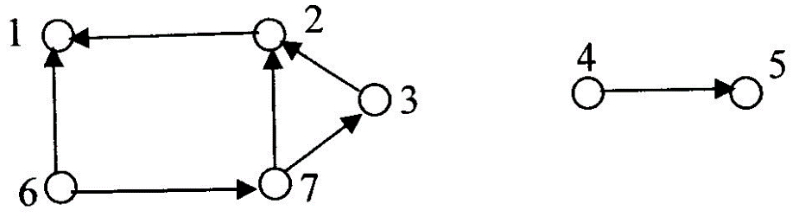
# NANYANG TECHNOLOGICAL UNIVERSITY
## School of Electrical & Electronic Engineering

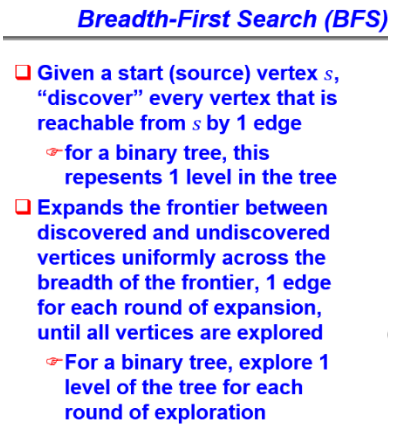
**<u>IE2108 Data Structures and Algorithms</u>**

**Tutorial No. 10 (Sem 1, AY2022-2023)**

1. Write an algorithm that prints the indegree and outdegree of every vertex in a digraph, where the digraph is represented using adjacency lists. Show output for the figure below.

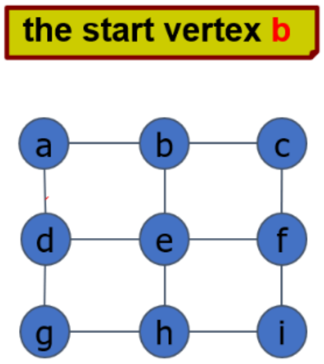


2. Trace the breadth-first search algorithm below on the graph below and show only the sequence of vertices visited:



***Breadth-First Search (BFS)***

- ❑ Given a start (source) vertex *s*, "discover" every vertex that is reachable from *s* by 1 edge
  - ☞for a binary tree, this repesents 1 level in the tree
- ❑ Expands the frontier between discovered and undiscovered vertices uniformly across the breadth of the frontier, 1 edge for each round of expansion, until all vertices are explored
  - ☞For a binary tree, explore 1 level of the tree for each round of exploration



3. Obtain the adjacency list for the above graph.

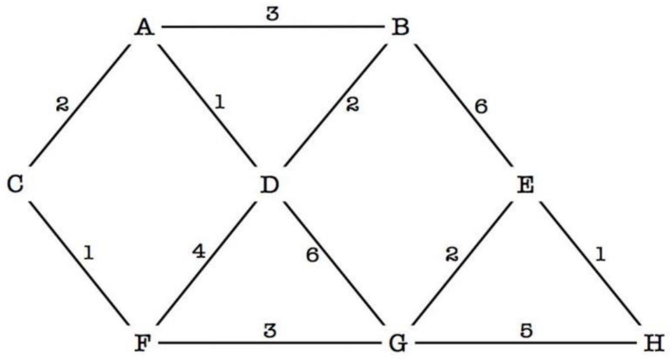
4. Trace the breadth-first search pseudocode below on the graph above and show the vertex visited and the queue at each iteration.

```
bfs (adj, s) {
       n = adj.last
       for i = 1 to n
            visit[i] = false
       q.enqueue(s)
       while (!q.empty()) {
            v = q.front()
            ref = adj[v]
            while (ref != null) {
                 if (!visit[ref.data]) {
                      q.enqueue(ref.data)
                 }
                 ref = ref.next
            }
            q.dequeue()
```

```
                visit[v] = true
            }

        }
```

5. You are given the following graph:



Do a breadth-first search of the graph, starting from node A. There are several possible orders that a breadth-first search could choose. Show two of these orders.