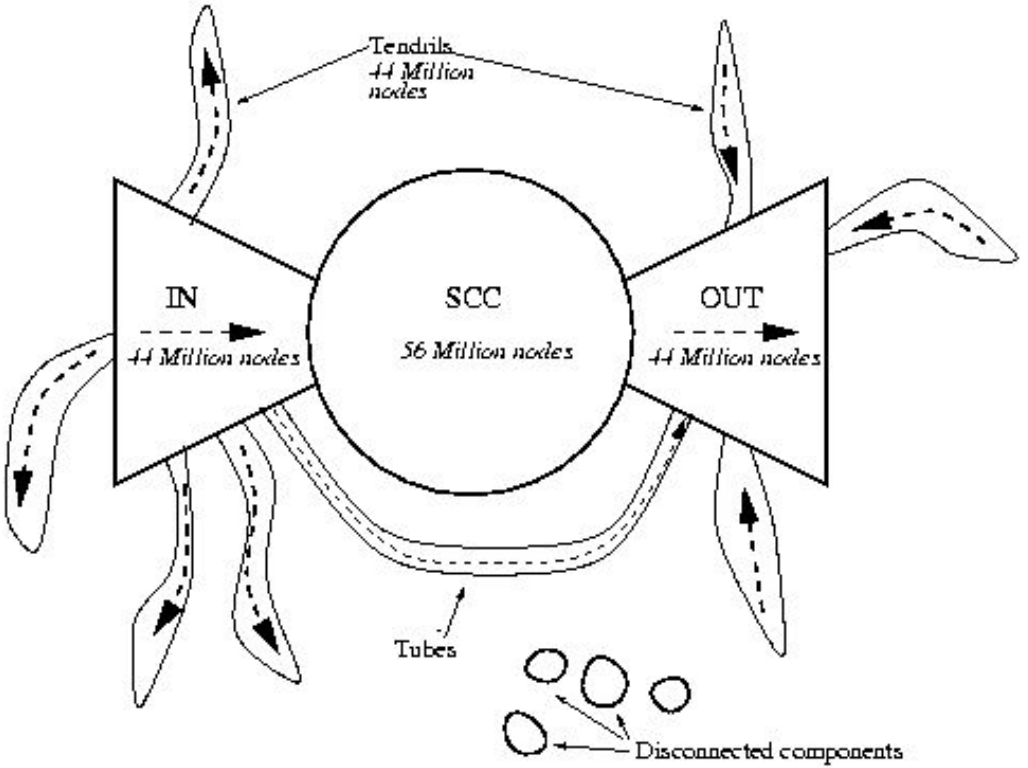
COMP20007 Design of Algorithms: Week 4

Student reps



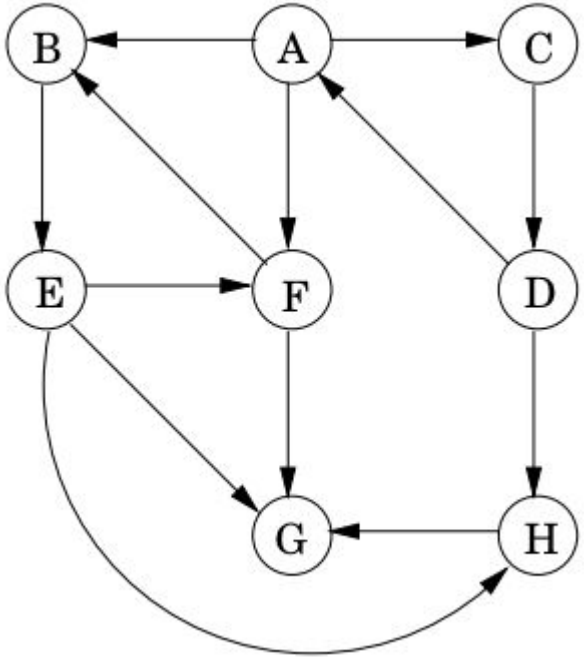
DFS algorithm

- Basic algorithm
- Adapt DFS to compute cyclicity, connectedness, colouring
- implementations

```
traverse(v):
    s = stack
    s.push(v)
    while s not empty:
        v = s.pop()
        mark v visited
        foreach w adjacent to v:
            if w not visited:
                s.push(w)
```

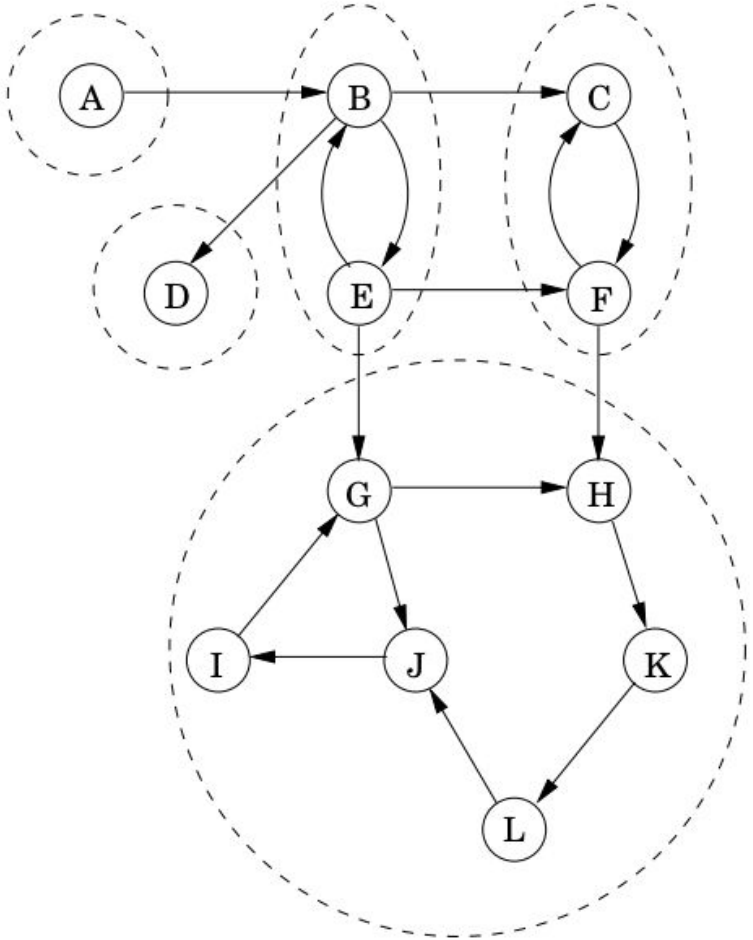
DFS Traversal

- Pre and post numbers
- Checking for cyclicity
- Connectivity
- SCC



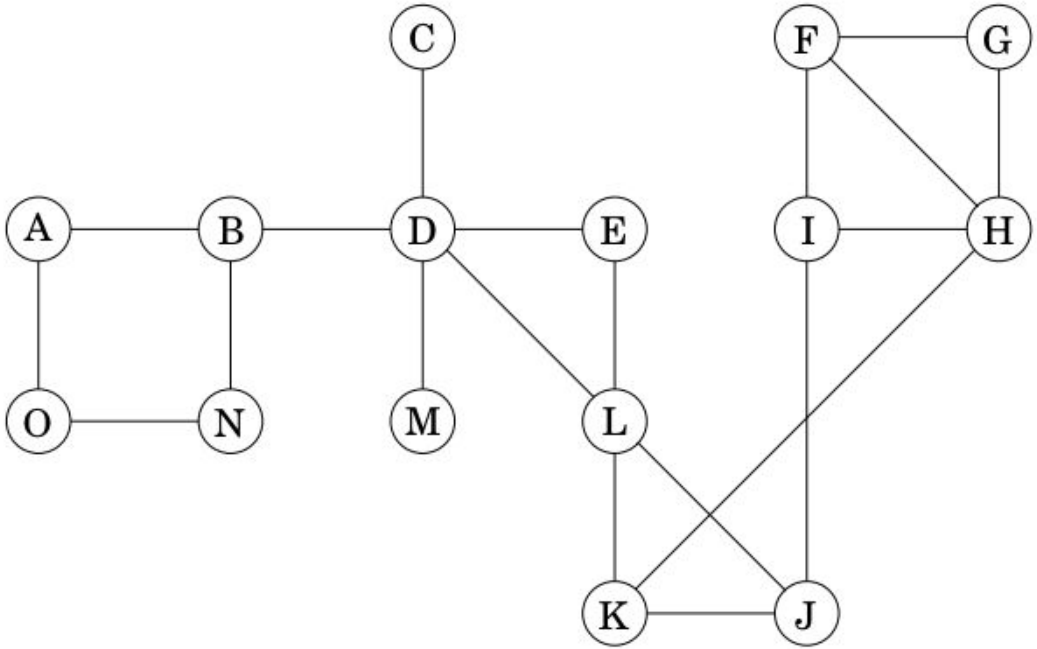
Strongly Connected components

- Source and sink SCCs

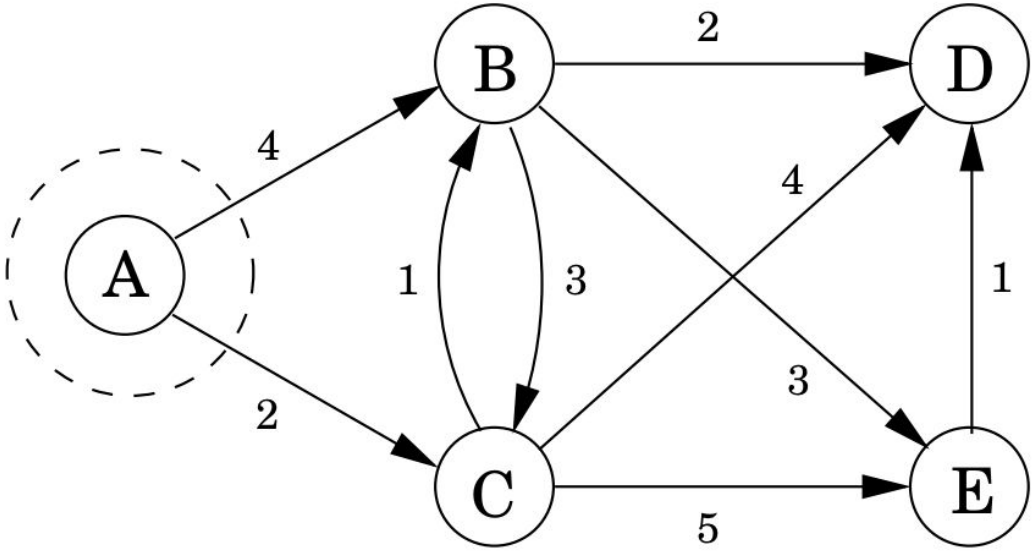


Distances in graphs

- BFS algorithm
- Definition of path



Dijkstra's Algorithm



Taking stock

Things you knew before we started:

- Array, Linked List, Binary Search Tree, Hash Table
- Stack, Queue, Priority Queue
- Sorting algorithms

New things covered in the first four weeks:

- Divide and conquer
- Formal analysis: recurrence relations, Master Theorem, big O
- Graphs: definition, properties, representations, DFS, BFS, topological sort
- Multi-module programs

Substantial reading over the break: DPV 3.1-4, 4.1-4