Operations on graph

- Graph has less structure than tree
  - Need more general operations
- Traversal / search
  - Find node by following edges from start node
  - Traversal methods
    - Depth-first traversal
    - Breadth-first traversal
Depth-first traversal

• How to implement depth-first traversal?

  Recursive implementation
  – Visit nodes following edges
  – Termination condition: node already visited

• Need process to track which nodes visited
  – Boolean variable in node

• Recursive call

  public void depthFirstTraversal(Vertex start) {
    System.out.print("DFS from "+start+": ");
    clearVisited();
    recursiveDFS(start);
    System.out.println();
  }
Depth-first traversal

- Recursion

```java
public void recursiveDFS(Vertex v) {
    if (v.visited)
        return;

    System.out.format("%3s ",v);
    v.visited = true;
    for (int i=0; i<activeVertices; i++)
        if (edges[v.graphIndex][i]==1) {
            recursiveDFS(vertices[i]);
        }
}
```

Breadth-first traversal

- How to implement breadth-first traversal?
Breadth-first traversal

- Need to keep track of children of nodes to be visited later
  - Queue / linked list maintains FIFO order
- Process
  - Add children of node into queue
  - Mark nodes as visited when entering in queue
  - Skip nodes that have been visited already

```java
public void breadthFirstTraversal(Vertex start) {
    System.out.print("BFS from " + start +": ");
    clearVisited();
    Queue<Vertex> vertexList = new LinkedList<Vertex>();
    vertexList.add(start);
    start.visited = true;
    while (!vertexList.isEmpty()) {
        Vertex current = vertexList.remove();
        System.out.format("%3s ",current);
        for (int i=0; i<activeVertices; i++) {
            if (edges[current.graphIndex][i]==1) {
                if (!vertices[i].visited) {
                    vertices[i].visited = true;
                    vertexList.add(vertices[i]);
                }
            }
        }
    }
    System.out.println();
}
```
Graph example

- (Hypothetical) airline connections between airports in Northeast
- Adjacency matrix:

<table>
<thead>
<tr>
<th></th>
<th>ALB</th>
<th>BDL</th>
<th>LGA</th>
<th>JFK</th>
<th>EWR</th>
<th>PVD</th>
<th>BOS</th>
<th>MHT</th>
<th>BTV</th>
<th>SWF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALB</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BDL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>LGA</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>JFK</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EWR</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PVD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BOS</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MHT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BTV</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SWF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Graph traversal example

- Traversal examples
  DFS from BDL: BDL EWR MHT PVD JFK ALB LGA BTV BOS HPN SWF
  BFS from BDL: BDL EWR BOS MHT HPN SWF PVD JFK BTV ALB LGA
  DFS from ALB: ALB LGA BTV PVD JFK MHT EWR BDL BOS HPN SWF
  BFS from ALB: ALB LGA JFK BTV PVD MHT EWR BDL BOS HPN SWF
Next Steps

• Discussions tomorrow
• Lecture on Friday
• Extra TA office hours on Friday