Graphs

To study all of these structures:
1. A common vocabulary
2. Graph implementations
3. Graph traversals
4. Graph algorithms
Graph ADT

Data:
- Vertices
- Edges
- Some data structure maintaining the structure between vertices and edges.

Functions:
- insertVertex(K key);
- insertEdge(Vertex v1, Vertex v2, K key);
- removeVertex(Vertex v);
- removeEdge(Vertex v1, Vertex v2);
- incidentEdges(Vertex v);
- areAdjacent(Vertex v1, Vertex v2);
- origin(Edge e);
- destination(Edge e);
Graph Implementation: Edge List

insertVertex(K key);
removeVertex(Vertex v);
areAdjacent(Vertex v1, Vertex v2);
incidentEdges(Vertex v);
Graph Implementation: Adjacency Matrix

- `insertVertex(K key)`
- `removeVertex(Vertex v)`
- `areAdjacent(Vertex v1, Vertex v2)`
- `incidentEdges(Vertex v)`

```
<table>
<thead>
<tr>
<th></th>
<th>u</th>
<th>v</th>
<th>w</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>u</td>
<td>v</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z</td>
<td>w</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Diagram:
- Vertices: u, v, w, z
- Edges: a, b, c, d
Graph Implementation: Edge List

- `insertVertex(K key);`
- `removeVertex(Vertex v);`
- `areAdjacent(Vertex v1, Vertex v2);`
- `incidentEdges(Vertex v);`

Diagram:
- Vertices: U, V, W, Z
- Edges: (U, V), (V, W), (U, W), (W, Z), (U, Z)

Edge List:
- (u, v, a)
- (v, w, b)
- (u, w, c)
- (w, z, d)
<table>
<thead>
<tr>
<th></th>
<th>Edge List</th>
<th>Adjacency Matrix</th>
<th>Adjacency List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressed as $O(f)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Space</td>
<td>$n+m$</td>
<td>$n+m$</td>
<td>$n^2$</td>
</tr>
<tr>
<td>insertVertex($v$)</td>
<td>1</td>
<td>$n$</td>
<td>1</td>
</tr>
<tr>
<td>removeVertex($v$)</td>
<td>$m$</td>
<td>$n$</td>
<td>$\text{deg}(v)$</td>
</tr>
<tr>
<td>insertEdge($v$, $w$, $k$)</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>removeEdge($v$, $w$)</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>incidentEdges($v$)</td>
<td>$m$</td>
<td>$n$</td>
<td>$\text{deg}(v)$</td>
</tr>
<tr>
<td>areAdjacent($v$, $w$)</td>
<td>$m$</td>
<td>1</td>
<td>$\min(\text{deg}(v), \text{deg}(w))$</td>
</tr>
</tbody>
</table>
Traversals:

Objective: Visit every vertex and every edge in the graph.

Purpose: Search for interesting sub-structures in the graph.

We’ve seen traversal before ....but it’s different:

- Ordered
- Obvious Start
- 
- 
- 
-
Traversal: BFS
CS 225 – Things To Be Doing

Exam 11 (theory) starts Monday after break
More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP6: A one week reflection MP!
*Due: Friday, Nov. 17 at 11:59pm*

MP7: The final MP!
*Extra Credit (+14): Monday, Dec. 4 at 11:59pm*
*Due: Monday, Dec. 11 at 11:59pm*

Lab: lab_dict this week
*Due: Wednesday, Nov. 29 @ 7pm  (Before the first lab after break!)*

No POTDs over break
*Worth +1 Extra Credit /problem (up to +40 total)*