Graph Traversals
And Reachability

Stephen Freund
CSCI 136

Last Time

Adjacency Matrix:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>30</td>
<td>-</td>
</tr>
</tbody>
</table>

Adjacency List:

A: [A->B,10] [A->C,20]
B: 
C: [C->B,30]

Depth-First Search

```java
void reachableFrom(Graph<V,E> g, V src) {
    if (!g.visited(src)) {
        g.visit(src);
        Iterator<V> neighbors = g.neighbors(src);
        while (neighbors.hasNext()) {
            reachableFrom(g, neighbors.next());
        }
    }
}
```
Shortest Path (No Weights on Edges)
**Unicycle Factory**

- Attach Seat to Frame
- Weld Frame
- Paint Frame
- Attach Wheel to Frame
- Attach Pedals to Wheel
- Inflate Tire

**Todo Stack**

- Finished Unicycle

- Inflate Tire
- Attach Wheel
- Attach Pedals
- Paint Frame
- Weld Frame
Finished Unicycle

Attach Wheel

Attach Pedals

Inflate Tire

Paint Frame

Weld Frame

Attach Seat

Attach Pedals

Paint Frame

Weld Frame

Finished Unicycle

todo stack
1. Attach Pedals
2. Weld Frame
3. Paint Frame
4. Attach Wheel
5. Inflate Tire
6. Attach Seat
7. Finished Unicycle