YOU KNOW THIS METAL RECTANGLE FULL OF LITTLE LIGHTS?

Yeah.

I SPEND MOST OF MY LIFE PRESSING BUTTONS TO MAKE THE PATTERN OF LIGHTS CHANGE HOWEVER I WANT. SOUNDS GOOD.

BUT TODAY, THE PATTERN OF LIGHTS IS ALL WRONG! OH GOD! TRY PRESSING MORE BUTTONS!

IT'S NOT HELPING!
Section 6: HW6

Slides by Alex Mariakakis

with material from Krysta Yousoufian, Mike Ernst, Kellen Donohue
Handling Expensive RIs

• **Problem:** a thorough `checkRep()` may take a while to execute; if it is called every time the graph is modified, your code may fail the 30 second timeout per test

• **Simple solution:** use a “debug flag” boolean to turn `checkRep()` on or off

• **Fancy solution:** make multiple `checkRep()` methods of different complexity and switch between them using an enum
Can I reach B from A?
Breadth-First Search (BFS)

- Often used for discovering connectivity
- Calculates the shortest path if and only if all edges have same positive or no weight
- Depth-first search (DFS) is commonly mentioned with BFS
  - BFS looks “wide”, DFS looks “deep”
  - Can also be used for discovery, but not the shortest path
BFS Pseudocode

```java
public boolean find(Node start, Node end) {
    put start node in a queue
    while (queue is not empty) {
        pop node N off queue
        if (N is goal)
            return true;
        else {
            for each node O that is child of N
                push O onto queue
        }
    }
    return false;
}
```
Breadth-First Search

Q: <>
Q: <A>
Q: <>
Q: <B>
Q: <B, C>
DONE
Breadth-First Search with Cycle

Q: <>
Q: <A>
Q: <>
Q: <B>
Q: <>
Q: <C>
Q: <>
Q: <A>
NEVER DONE
BFS Pseudocode

```java
public boolean find(Node start, Node end) {
    put start node in a queue
    while (queue is not empty) {
        pop node N off queue
        if (N is goal)
            return true;
        else {
            for each node O that is child of N
                push O onto queue
        }
    }
    return false;
}
```

Mark the node as visited!
Breadth-First Search

Q: <>
Breadth-First Search

Q: <>
Q: <A>
Breadth-First Search
Breadth-First Search

Q: <>
Q: <A>
Q: <>
Q: <C>
Breadth-First Search

Q: <>
Q: <A>
Q: <>
Q: <C>
Q: <C, D>
Breadth-First Search

Q: <>
Q: <A>
Q: <>
Q: <C>
Q: <C, D>
Q: <D>
Breadth-First Search

Q: <>
Q: <A>
Q: <>
Q: <C>
Q: <C, D>
Q: <D>
Q: <D, E>
Breadth-First Search

Q: <>
Q: <A>
Q: <>
Q: <C>
Q: <C, D>
Q: <D>
Q: <D, E>
Q: <E>
Breadth-First Search

Q: <>
Q: <A>
Q: <>
Q: <C>
Q: <C, D>
Q: <D>
Q: <D, E>
Q: <E>
DONE
Shortest Paths with BFS

From Node B

<table>
<thead>
<tr>
<th>Destination</th>
<th>Path</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>&lt;B,A&gt;</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>&lt;B&gt;</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>&lt;B,A,C&gt;</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>&lt;B,D&gt;</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>&lt;B,D,E&gt;</td>
<td>2</td>
</tr>
</tbody>
</table>
Shortest Paths with Weights

Paths are not the same!

From Node B

<table>
<thead>
<tr>
<th>Destination</th>
<th>Path</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>&lt;B,A&gt;</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>&lt;B&gt;</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>&lt;B,A,C&gt;</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>&lt;B,A,C,D&gt;</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>&lt;B,A,C,E&gt;</td>
<td>7</td>
</tr>
</tbody>
</table>
Demo
Parsing the Marvel data