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!
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 *(Do this!)*

- **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

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;
}
Q: <>
Q: <A>
Q: <>
Q: <B>
Q: <B, C>
DONE

Starting at node A
BREADTH-FIRST SEARCH WITH CYCLE

Starting at node A

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

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;
}
Q: <>
BREADTH-FIRST SEARCH

Q: <>
Q: <A>

Starting at node A
Destination node is E
BREADTH-FIRST SEARCH

Q: <>
Q: <A>
Q: <>
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>
What about B?
BREADTH-FIRST SEARCH

Q: <>
Q: <A>
Q: <>
Q: <C>
Q: <C, D>
Q: <D>
Q: <D, E>
Q: <E>
DONE

No path from B to A, so not included
What was the starting node?
What was the starting node? Node B
### SHORTEST PATHS WITH WEIGHTS

**Nodes:** A, B, C, D, E

**Starting 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>

Paths are not the same!
Will Discuss Next Week for HW7!
HW6 OVERVIEW

- Look at marvel.tsv file
- Parsing of file done for you, look at MarvelParser.java
- Fill up your graph (may need to make changes to your Graph ADT)
- Find shortest path between two characters through the different comic books that they appear in with other characters
- Testing!
Demo
HW6