The Project 2 DiGraph interface

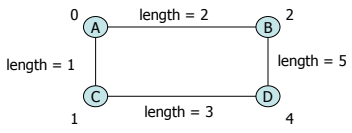
A quick introduction to some simple usage

DiGraph type

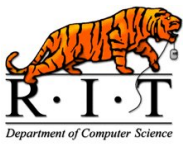
- Provides an interface definition for directed graphs
- Can also be used to support undirected graphs
  - How?
- JavaDocs are available online

A sample problem

- Consider the following graph:
  - Edges are labeled with the lengths between points
  - Vertices are labeled with the distances from "A"



- What calls need to be made on a DiGraph object to build this graph, given this data?



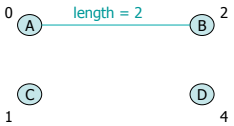
No peeking....

Creating the vertices

```
theGraph.addVertex( "A", new Integer(0) );
theGraph.addVertex( "B", new Integer(2) );
theGraph.addVertex( "C", new Integer(1) );
theGraph.addVertex( "D", new Integer(4) );
```

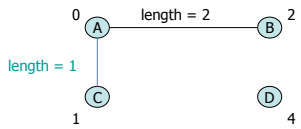
Connecting A&B

```
theGraph.addEdge( "A", "B", new Integer( 2 ) );
theGraph.addEdge( "B", "A", new Integer( 2 ) );
```



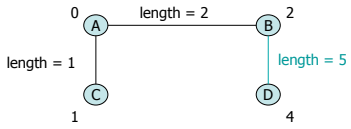
Connecting A&C

```
theGraph.addEdge( "A", "C", new Integer( 1 ) );
theGraph.addEdge( "C", "A", new Integer( 1 ) );
```



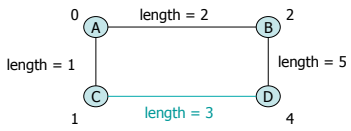
Connecting B&D

```
theGraph.addEdge( "B", "D", new Integer( 5 ) );
theGraph.addEdge( "D", "B", new Integer( 5 ) );
```



Connecting C&D

```
theGraph.addEdge( "C", "D", new Integer( 3 ) );
theGraph.addEdge( "D", "C", new Integer( 3 ) );
```



Putting it all together

```
theGraph.addVertex( "A", new Integer(0) );
theGraph.addVertex( "B", new Integer(2) );
theGraph.addVertex( "C", new Integer(1) );
theGraph.addVertex( "D", new Integer(4) );

theGraph.addEdge( "A", "B", new Integer( 2 ) );
theGraph.addEdge( "B", "A", new Integer( 2 ) );

theGraph.addEdge( "A", "C", new Integer( 1 ) );
theGraph.addEdge( "C", "A", new Integer( 1 ) );

theGraph.addEdge( "B", "D", new Integer( 5 ) );
theGraph.addEdge( "D", "B", new Integer( 5 ) );

theGraph.addEdge( "C", "D", new Integer( 3 ) );
theGraph.addEdge( "D", "C", new Integer( 3 ) );
```