Graph Coverage
In-class exercise

Dr. Brittany Johnson-Matthews
(Dr. B for short)

https://go.gmu.edu/SWE637
Adapted from slides by Jeff Offutt and Bob Kurtz
Graph Coverage Exercise 1

Consider the following graph:

\[
N = \{ 1, 2, 3, 4, 5, 6 \}
\]

\[
NO = \{ 1 \}
\]

\[
Nf = \{ 6 \}
\]

\[
E = \{ (1,2), (2,3), (3,4), (3,5), (4,5), (5,2), (2,6) \}
\]

\[
def(1) = def(4) = use(3) = use(5) = use(6) = \{ x \}
\]
Graph Coverage Exercise 1

1. Draw the graph and annotate it with defs and uses

2. List all the DU-paths with respect to x, including any that are subpaths of other DU-paths

3. Given the following test paths, which DU-paths are toured (directly, with a detour, or with a side-trip) by each test path?
   - t1 = [ 1, 2, 3, 5, 2, 6 ]
   - t2 = [ 1, 2, 3, 4, 5, 2, 6 ]

4. List a minimal test set (that is, a set of test paths) that satisfies all defs coverage using direct tours

5. List a minimal test set (that is, a set of test paths) that satisfies all du-paths coverage using direct tours
Graph Coverage Exercise 1

Draw the graph.

Diagram:

1. def(x)
2. 
3. use(x)
4. def(x)
5. use(x)
6. use(x)
Graph Coverage Exercise 1

List all the $DU$-paths with respect to $x$, including any that are subpaths of other $DU$-paths

$[1,2,3], [1,2,3,5], [1,2,6], [4,5], [4,5,2,3], [4,5,2,6]$
### Graph Coverage Exercise 1

Which DU-paths are toured by each test path?

Mark “✓” for direct tour, “D” for detour, “S” for sidetrip

$t_1 = [ 1, 2, 3, 5, 2, 6 ]$

$t_2 = [ 1, 2, 3, 4, 5, 2, 6 ]$

<table>
<thead>
<tr>
<th>DU-Path</th>
<th>$t_1$</th>
<th>$t_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,2,3]</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>[1,2,3,5]</td>
<td>✓</td>
<td>✔</td>
</tr>
<tr>
<td>[1,2,6]</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>[4,5]</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>[4,5,2,3]</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>[4,5,2,6]</td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>
Which DU-paths are toured by each test path?

Mark “✓” for direct tour, “D” for detour, “S” for sidetrip

$t_1 = [1, 2, 3, 5, 2, 6]$  
$t_2 = [1, 2, 3, 4, 5, 2, 6]$ 

<table>
<thead>
<tr>
<th>DU-Path</th>
<th>$t_1$</th>
<th>$t_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,2,3]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[1,2,3,5]</td>
<td>✓</td>
<td>D</td>
</tr>
<tr>
<td>[1,2,6]</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>[4,5]</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>[4,5,2,3]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[4,5,2,6]</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Graph Coverage Exercise 1

List a minimal test set (that is, a set of test paths) that satisfies all defs coverage using direct tours:
Graph Coverage Exercise 1

List a minimal test set (that is, a set of test paths) that satisfies all defs coverage using direct tours:

\{ [1,2,3,4,5,2,6] \}
Graph Coverage Exercise 1

List a minimal test set (that is, a set of test paths) that satisfies all du-paths coverage using direct tours:
Graph Coverage Exercise 1

List a minimal test set (that is, a set of test paths) that satisfies all du-paths coverage using direct tours:

\[
\{ [1,2,3,5,2,6], \\
[1,2,3,4,5,2,6], \\
[1,2,3,4,5,2,3,5,2,6], \\
[1,2,6] \}
\]
END OF EXERCISE 1
public static <T extends Comparable<? super T>>
T min (List<? extends T> list) {
    Iterator<? extends T> itr = list.iterator();
    if (!itr.hasNext()) {
        throw new IllegalArgumentException("min: empty list");
    }
    T result = itr.next();
    if (result == null)
        throw new NullPointerException();
    while (itr.hasNext()) {
        T comp = itr.next();
        if (comp.compareTo(result) < 0) { // throws NPE, CCE
            result = comp;
        }
    }
    return result;
}
Graph Coverage Exercise 2

1. Draw the graph
2. Develop node coverage test requirements
3. Develop edge coverage test requirements
4. Develop edge-pair coverage test requirements
5. Develop prime path coverage test requirements
Graph Coverage Exercise 2

Draw the graph.
Graph Coverage Exercise 2

Draw the graph.

```
itr=...
hasNext==T
```

```
result=...
result!=null
```

```
hasNext==T
compareTo>=0
compareTo<0
result=...
```

```
hasNext==F
```

```
throw IAE
```

```
result==null
```

```
throw NPE
```

```
throw NPE
```

```
throw CCE
```

```
result==null
```

```
hasNext==F
```

```
return
```
Graph Coverage Exercise 2

Develop node coverage test requirements:
Graph Coverage Exercise 2

Develop node coverage test requirements:

\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}
Graph Coverage Exercise 2

Develop edge coverage test requirements:
Graph Coverage Exercise 2

Develop edge coverage test requirements:

\{ (1,2), (1,3), (2,4), \\
(2,5), (4,6), (4,10), \\
(6,4), (6,7), (6,8), \\
(6,9), (7,4) \}
Graph Coverage Exercise 2

Develop edge-pair coverage test requirements:
Graph Coverage Exercise 2

Develop edge-pair coverage test requirements:

\{ (1,2,4), (1,2,5), (1,3), (2,4,6), (2,4,10), (4,6,4), (4,6,7), (4,6,8), (4,6,9), (6,4,6), (6,4,10), (6,7,4), (7,4,6), (7,4,10) \}
Graph Coverage Exercise 2

Develop prime path coverage test requirements:
Graph Coverage Exercise 2

Develop prime path coverage test requirements:

\{ (1,2,4,6,7),
    (1,2,5), (1,3),
    (1,2,4,10),
    (1,2,4,6,8),
    (1,2,4,6,9),
    (4,6,4), (4,6,7,4),
    (6,4,6), (6,4,10),
    (6,7,4,6), (6,7,4,10),
    (7,4,6,7),
    (7,4,6,8),
    (7,4,6,9) \}
END OF EXERCISE 2