Analysis

Problem: Used a Map for the nodes, but I couldn't distinguish between a type of Resource or type of Process without getting a collision, so any values that are equal will not both be added. For example, the line:

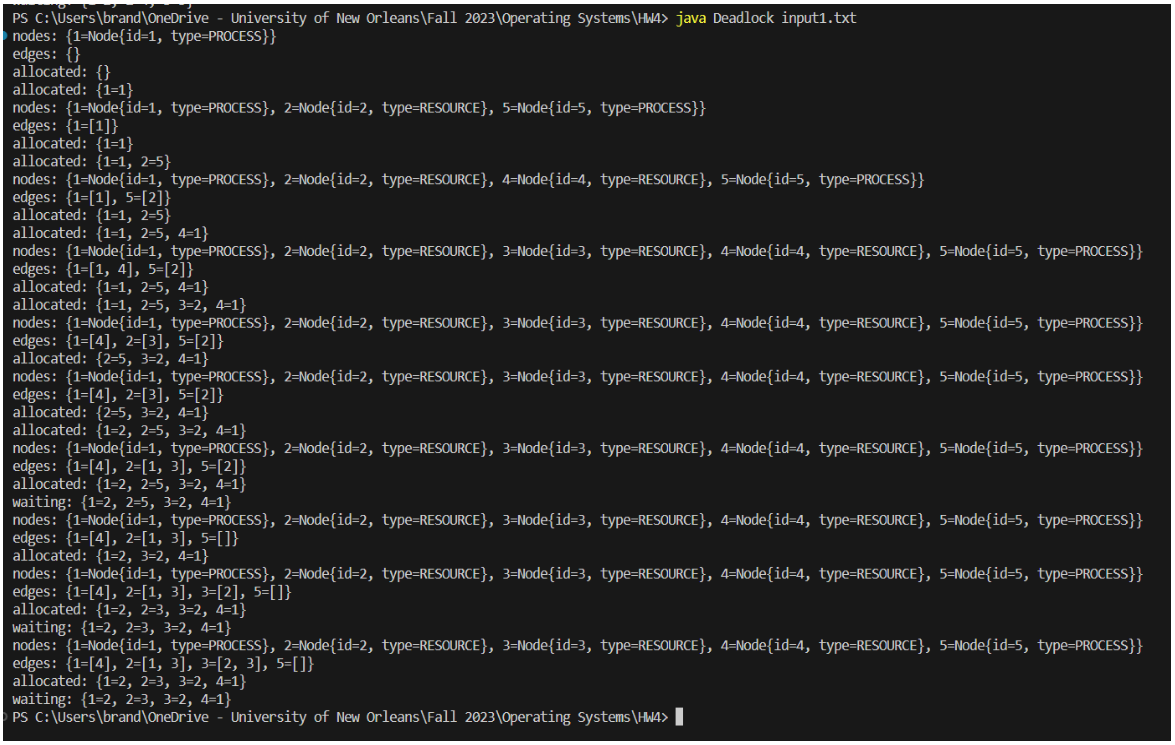
1 W 1

(Resource 1 is allocated to Process 1)

Will only create a node for the process instead of both of them.

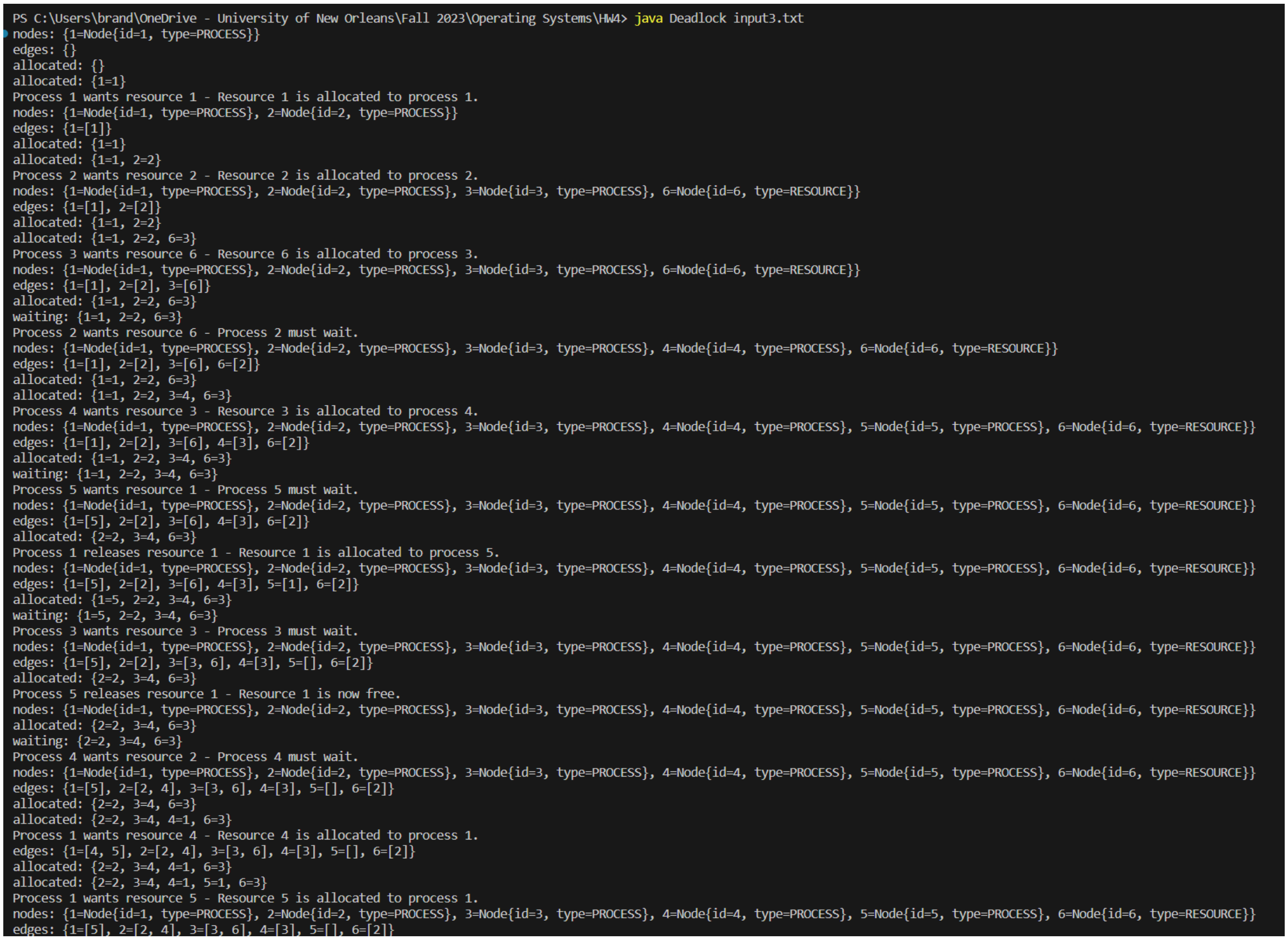
I believe all the other code works fine though. I spent a lot of time messing with different data structures (like an entire week) to be used as the node (Linked Hashmap, LinkedList, Stack, etc.) and I couldn't get them to work with my DFS cycle detection or my deadlockDetection functions without breaking those. I also tried doing it again from scratch with addNode and addEdge but I couldn't figure out how to distinguish a resource/process node and a request/waiting edge, which made it hard to determine things like waiting processes and the DFS.

Never worked with graphs before so this was a rough one to figure out. Not sure if I was overcomplicating this or undercomplicating it.



```
PS C:\Users\brand\OneDrive - University of New Orleans\Fall 2023\Operating Systems\HW4> java Deadlock input1.txt
nodes: {1=Node{id=1, type=PROCESS}}
edges: {}
allocated: {}
allocated: {1=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[1]}
allocated: {1=1}
allocated: {1=1, 2=5}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[1], 5=[2]}
allocated: {1=1, 2=5}
allocated: {1=1, 2=5, 4=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 3=Node{id=3, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[1, 4], 5=[2]}
allocated: {1=1, 2=5, 4=1}
allocated: {1=1, 2=5, 3=2, 4=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 3=Node{id=3, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[4], 2=[3], 5=[2]}
allocated: {2=5, 3=2, 4=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 3=Node{id=3, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[4], 2=[3], 5=[2]}
allocated: {2=5, 3=2, 4=1}
allocated: {1=2, 2=5, 3=2, 4=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 3=Node{id=3, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[4], 2=[1, 3], 5=[2]}
allocated: {1=2, 2=5, 3=2, 4=1}
waiting: {1=2, 2=5, 3=2, 4=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 3=Node{id=3, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[4], 2=[1, 3], 5=[]}
allocated: {1=2, 3=2, 4=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 3=Node{id=3, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[4], 2=[1, 3], 3=[2], 5=[]}
allocated: {1=2, 2=3, 3=2, 4=1}
waiting: {1=2, 2=3, 3=2, 4=1}
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=RESOURCE}, 3=Node{id=3, type=RESOURCE}, 4=Node{id=4, type=RESOURCE}, 5=Node{id=5, type=PROCESS}}
edges: {1=[4], 2=[1, 3], 3=[2, 3], 5=[]}
allocated: {1=2, 2=3, 3=2, 4=1}
waiting: {1=2, 2=3, 3=2, 4=1}
PS C:\Users\brand\OneDrive - University of New Orleans\Fall 2023\Operating Systems\HW4>
```

Example w/ input text 1



```
PS C:\Users\brand\OneDrive - University of New Orleans\Fall 2023\Operating Systems\HW4> java Deadlock input3.txt
nodes: {1=Node{id=1, type=PROCESS}}
edges: {}
allocated: {}
allocated: {1=1}
Process 1 wants resource 1 - Resource 1 is allocated to process 1.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}}
edges: {1=[1]}
allocated: {1=1}
allocated: {1=1, 2=2}
Process 2 wants resource 2 - Resource 2 is allocated to process 2.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2]}
allocated: {1=1, 2=2}
allocated: {1=1, 2=2, 6=3}
Process 3 wants resource 6 - Resource 6 is allocated to process 3.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2], 3=[6]}
allocated: {1=1, 2=2, 6=3}
waiting: {1=1, 2=2, 6=3}
Process 2 wants resource 6 - Process 2 must wait.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2], 3=[6], 6=[2]}
allocated: {1=1, 2=2, 6=3}
allocated: {1=1, 2=2, 3=4, 6=3}
Process 4 wants resource 3 - Resource 3 is allocated to process 4.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2], 3=[6], 4=[3], 6=[2]}
allocated: {1=1, 2=2, 3=4, 6=3}
waiting: {1=1, 2=2, 3=4, 6=3}
Process 5 wants resource 1 - Process 5 must wait.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[5], 2=[2], 3=[6], 4=[3], 6=[2]}
allocated: {2=2, 3=4, 6=3}
Process 1 releases resource 1 - Resource 1 is allocated to process 5.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[5], 2=[2], 3=[6], 4=[3], 5=[1], 6=[2]}
allocated: {1=5, 2=2, 3=4, 6=3}
waiting: {1=5, 2=2, 3=4, 6=3}
Process 3 wants resource 3 - Process 3 must wait.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[5], 2=[2], 3=[3, 6], 4=[3], 5=[], 6=[2]}
allocated: {2=2, 3=4, 6=3}
Process 5 releases resource 1 - Resource 1 is now free.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
allocated: {2=2, 3=4, 6=3}
waiting: {2=2, 3=4, 6=3}
Process 4 wants resource 2 - Process 4 must wait.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[5], 2=[2, 4], 3=[3, 6], 4=[3], 5=[], 6=[2]}
allocated: {2=2, 3=4, 6=3}
allocated: {2=2, 3=4, 4=1, 6=3}
Process 1 wants resource 4 - Resource 4 is allocated to process 1.
edges: {1=[4, 5], 2=[2, 4], 3=[3, 6], 4=[3], 5=[], 6=[2]}
allocated: {2=2, 3=4, 4=1, 6=3}
allocated: {2=2, 3=4, 4=1, 5=1, 6=3}
Process 1 wants resource 5 - Resource 5 is allocated to process 1.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS}, 4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[5], 2=[2, 4], 3=[3, 6], 4=[3], 5=[], 6=[2]}
```

Another example w/ input txt 3 from the assignment sheet

PS C:\Users\brand\OneDrive - University of New Orleans\Fall 2023\Operating Systems\HW4> java Deadlock input3.txt

nodes: {1=Node{id=1, type=PROCESS}}

edges: {}

allocated: {}

allocated: {1=1}

Process 1 wants resource 1 - Resource 1 is allocated to process 1.

nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}}

edges: {1=[1]}

allocated: {1=1}

```
allocated: {1=1, 2=2}
Process 2 wants resource 2 - Resource 2 is allocated to process 2.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2]}
allocated: {1=1, 2=2}
allocated: {1=1, 2=2, 6=3}
Process 3 wants resource 6 - Resource 6 is allocated to process 3.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2], 3=[6]}
allocated: {1=1, 2=2, 6=3}
waiting: {1=1, 2=2, 6=3}
Process 2 wants resource 6 - Process 2 must wait.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2], 3=[6], 6=[2]}
allocated: {1=1, 2=2, 6=3}
allocated: {1=1, 2=2, 3=4, 6=3}
Process 4 wants resource 3 - Resource 3 is allocated to process 4.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[1], 2=[2], 3=[6], 4=[3], 6=[2]}
allocated: {1=1, 2=2, 3=4, 6=3}
waiting: {1=1, 2=2, 3=4, 6=3}
Process 5 wants resource 1 - Process 5 must wait.
nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}
edges: {1=[5], 2=[2], 3=[6], 4=[3], 6=[2]}
allocated: {2=2, 3=4, 6=3}
```

```
Process 1 releases resource 1 - Resource 1 is allocated to process 5.

nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}

edges: {1=[5], 2=[2], 3=[6], 4=[3], 5=[1], 6=[2]}

allocated: {1=5, 2=2, 3=4, 6=3}

waiting: {1=5, 2=2, 3=4, 6=3}

Process 3 wants resource 3 - Process 3 must wait.

nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}

edges: {1=[5], 2=[2], 3=[3, 6], 4=[3], 5=[], 6=[2]}

allocated: {2=2, 3=4, 6=3}

Process 5 releases resource 1 - Resource 1 is now free.

nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}

allocated: {2=2, 3=4, 6=3}

waiting: {2=2, 3=4, 6=3}

Process 4 wants resource 2 - Process 4 must wait.

nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}

edges: {1=[5], 2=[2, 4], 3=[3, 6], 4=[3], 5=[], 6=[2]}

allocated: {2=2, 3=4, 6=3}

allocated: {2=2, 3=4, 4=1, 6=3}

Process 1 wants resource 4 - Resource 4 is allocated to process 1.

edges: {1=[4, 5], 2=[2, 4], 3=[3, 6], 4=[3], 5=[], 6=[2]}

allocated: {2=2, 3=4, 4=1, 6=3}

allocated: {2=2, 3=4, 4=1, 5=1, 6=3}

Process 1 wants resource 5 - Resource 5 is allocated to process 1.

nodes: {1=Node{id=1, type=PROCESS}, 2=Node{id=2, type=PROCESS}, 3=Node{id=3, type=PROCESS},
4=Node{id=4, type=PROCESS}, 5=Node{id=5, type=PROCESS}, 6=Node{id=6, type=RESOURCE}}

edges: {1=[5], 2=[2, 4], 3=[3, 6], 4=[3], 5=[], 6=[2]}
```

```
allocated: {2=2, 3=4, 5=1, 6=3}

Process 1 releases resource 4 - Resource 4 is now free.

EXECUTION COMPLETED: No deadlock encountered.
```