L9 PROBLEM 2 (10/10 points)

Consider our representation of permutations of students in a line from Problem 1. In this case, we will consider a line of three students, Alice, Bob, and Carol (denoted A, B, and C). Using the Graph class created in the lecture, we can create a graph with the design chosen in Problem 1. To recap, vertices represent permutations of the students in line; edges connect two permutations if one can be made into the other by swapping two adjacent students.

We construct our graph by first adding the following nodes:

```
nodes = []
nodes.append(Node("ABC")) # nodes[0]
nodes.append(Node("ACB")) # nodes[1]
nodes.append(Node("BAC")) # nodes[2]
nodes.append(Node("BCA")) # nodes[3]
nodes.append(Node("CAB")) # nodes[4]
nodes.append(Node("CBA")) # nodes[5]

g = Graph()
for n in nodes:
    g.addNode(n)
```

Add the appropriate edges to the graph.

Hint: How to get started?

```
adj_mat = {0:[1,2], 1:[4], 2:[3], 3:[5], 4:[5], 5:[]}

for nid in adj_mat:
    for nbrid in adj_mat[nid]:
        g.addEdge(Edge(nodes[nid], nodes[nbrid]))
```

Correct

```
g.addEdge(Edge(nodes[0], nodes[1]))
g.addEdge(Edge(nodes[0], nodes[2]))
g.addEdge(Edge(nodes[1], nodes[4]))
g.addEdge(Edge(nodes[2], nodes[3]))
g.addEdge(Edge(nodes[3], nodes[5]))
g.addEdge(Edge(nodes[4], nodes[5]))

# or some variation on this. Obviously, in a Graph,
# g.addEdge(Edge(nodes[0], nodes[1])) functions just as well as
# g.addEdge(Edge(nodes[1], nodes[0])).
```

Test results

Hide output

CORRECT

Test: edges from node0

Output:

```
edges = g.childrenOf(nodes[0])
Printing out each edge...
ACB
BAC
```

Test: edges from node1

Output:

```
edges = g.childrenOf(nodes[1])
Printing out each edge...
ABC
CAB
```

Test: edges from node2

Output:

```
edges = g.childrenOf(nodes[2])
Printing out each edge...
ABC
BCA
```

Test: edges from node3

Output:

```
edges = g.childrenOf(nodes[3])
Printing out each edge...
BAC
CBA
```

Test: edges from node4

Output:

```
edges = g.childrenOf(nodes[4])
Printing out each edge...
ACB
CBA
```

Test: edges from node5

Output:

```
edges = g.childrenOf(nodes[5])
Printing out each edge...
BCA
CAB
```

Hide output

Check | Hide Answer

Show Discussion

New Post