CSE 250 Recitation

10/16-10/17 : Stacks, Queues, Graph Basics, and PA2

Stacks vs Queues

What does the following code do when MysterySequence is a Stack? Queue?

What are the relevant operations for each?

What are their runtimes for different backing data structures?

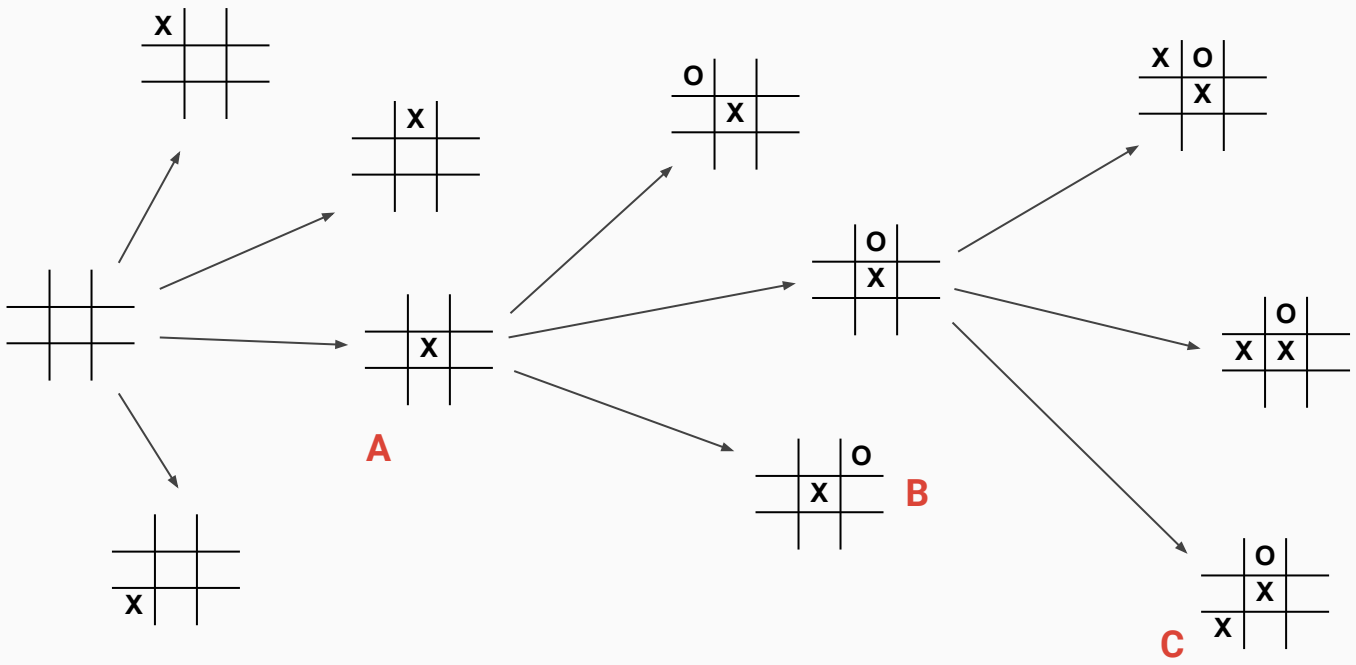
```
MysterySequence seq = new MysterySequence()
seq.addSomething("A")
seq.addSomething("B")
seq.addSomething("C")
seq.addSomething("D")
print(seq.removeSomething())
print(seq.removeSomething())
print(seq.removeSomething())
seq.addSomething("E")
print(seq.removeSomething())
seq.addSomething("F")
print(seq.removeSomething())
seq.addSomething("G")
seq.addSomething("H")
print(seq.removeSomething())
print(seq.removeSomething())
print(seq.removeSomething())
```

Graphs

How can the following things be represented as graphs? (ie. What would a vertex represent? What would an edge represent? What kind of work would we be using the graph for?)

- A street map of Buffalo
- Twitter
- Wikipedia
- A game of Tic-Tac-Toe

Tic Tac Toe Example



Note: This does not show all edges / vertices…

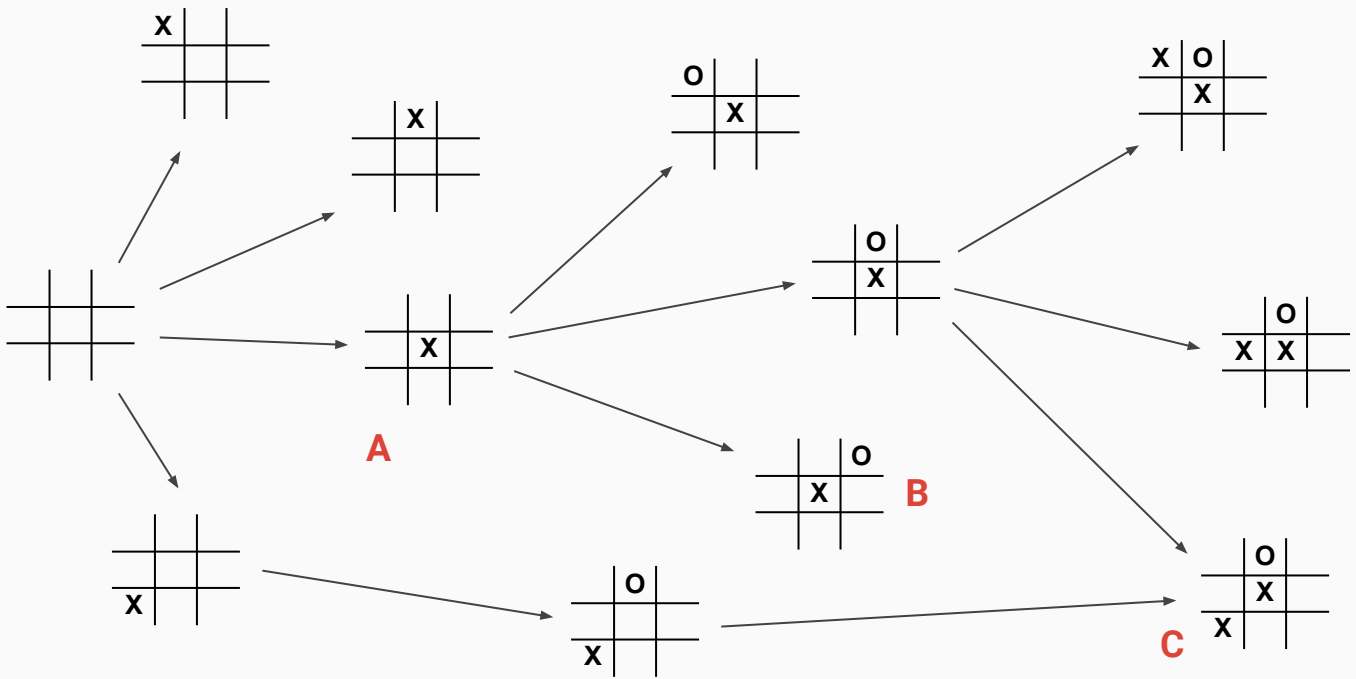
What is the out degree of the vertex for the empty board? What about the in degree?

What is the out degree of the vertex labeled A? B?

How many edges are in the full graph?

Is the in degree of every non-starting node 1?

Tic Tac Toe Example



Note: This does not show all edges / vertices…

What is the out degree of the vertex for the empty board? 9
What about the in degree? 0

What is the out degree of the vertex labeled A? B? 8, 7

How many edges are in the full graph? 9!

Is the in degree of every non-starting node 1? No ie C

PA2: Getting Started

- We will be starting this PA with another testing phase
- Remember, you don’t need to know how to implement an algorithm to start testing.
- The recommended way to start the testing is to draw a potential testing graph and see how different graph traversals can create different paths with the same starting node and ending node
  - BFS (Breadth First Search) will find the path that has the smallest number of edges possible
  - Dijkstra’s will find the path with smallest cost possible
- Now, with a partner or group try to come up with potential graphs you could use for testing

PA2: Getting Started

- What is the adjacency list for the graph to the right?
- What might make this graph good for testing?
  - (Hint: What do the different graph traversals return when used on the same graph)
- What are some things you can add to the graph to improve your tests?