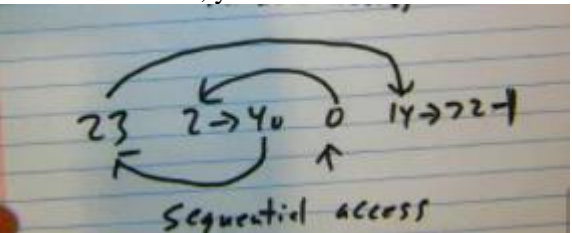


CSE143 Lecture Questions for Monday, 4/12/21

Question	Answer
<p>When you say that every list node has 2 things (the data and the reference), do you mean it only has those 2 things? Or can it store other things as well? Ok, makes sense, thanks!</p>	<p>At a minimum every list node will have some data and a reference to the next node. It can have more. I'd think of it as having extra data besides just one thing.</p>
<p>Why are fields in a ListNode public? Just to be able to read/write using the LinkedList? Sounds good</p>	<p>I'll discuss this in Wednesday's lecture.</p>
<p>Why is it important that we learn how to implement LinkedList, ArrayList.. Etc? These are all classes in java so why is it not enough to know how to use them? Cool! Thank you!</p>	<p>There are several answers to this. One is that you need to practice implementing data structures because you might want to implement other structures in the future. Another reason is that you understand the structure better when you have implemented it. We are trying to strike a balance between implementation and using a structure as a client.</p>
<p>Quite an out-of-topic question, but is a circular LinkedList possible? Can the 'last' element have a reference to the first one? I see. Very cool!</p>	<p>Yes, a circular list is possible and actually somewhat common. I had to make a choice about which kind of list to teach and I decided to teach the null-terminated version.</p>
<p>Can a node reference two things at once? !!! Interesting!</p>	<p>Not the nodes we have defined because they have just one link. But later we will learn about a structure called a binary tree where nodes refer to two other nodes.</p>
<p>How come at the start, you visualize linked lists like  this: where the numbers don't seem to be in any particular order, but later on they seem to just go left to right? So basically the earlier image is just meant to show that it doesn't have a particular order other than that they are linked sequentially? I see, thanks!</p>	<p>Putting the nodes in order from left to right is just meant to make it more convenient to see what's going on. In general, we don't have to think about where these things are stored in memory. Exactly. They can be in any order whatsoever in memory but the links establish a different kind of order that we can use when drawing a picture of them.</p>
<p>What's your favorite primitive type?</p>	<p>I think I'd go with int.</p>

<p>Is this a good analogy of how Linked Lists work?</p> <p>When I was in elementary school, sometimes getting into a properly ordered line (by last name) was important. So she would tell each of us to remember the person in front of you. Therefore when the line was formed it was ordered.</p>	<p>That's a reasonable analogy.</p>
<p>Why would you use type ListNode for the variable next? Shouldn't the type be int, since next is referring to the next element number, which is an int?</p> <p>How does type ListNode allow for access to that node?</p>	<p>We use references to keep track of what the next node in the list is. We can't use a simple int for that because it wouldn't allow us to access that node.</p> <p>It is storing a reference to that node. Java isn't specific about exactly what a reference is, but you could think of it as being the address in memory of the next node.</p>
<p>How can you use ListNode as a type when the class that creates it isn't built yet?</p>	<p>Making that work isn't trivial, but there is no reason that you can't record the type and insist that any type that has been mentioned eventually gets defined.</p>
<p>At 23:24 is that box drawn at the right side of "list" supposed to be "[" ← left and right bracket?</p> <p>So that the typed version would be "list["</p> <p>Got it, thanks</p>	<p>I'm not sure what you have in mind for the empty brackets. I'm drawing a picture of a node that has two fields, so empty brackets would not be appropriate. I draw a small box to the right of list to indicate that it can store a reference to a ListNode object.</p>
<p>Is there a way we can make a private class? I just thought that because making ListNode public seems like revealing implementation details (bad style).</p> <p>Thank you!</p>	<p>Eventually we will do that, but I want to keep it simpler for now. You can actually have a class defined inside another class, which is really the right thing to do with the node class. We'll see that towards the end of the course.</p>
<p>When does the Java Garbage Collector reclaim storage and how does it deem certain storage as garbage?</p> <p>Okay thanks!</p>	<p>The Java runtime system decides when to invoke the garbage collector, typically when it is running out of memory for creating new objects. An object is considered garbage if it is unreachable.</p>
<p>When will we be able to post our songs to Ed?</p> <p>Ok thanks!</p>	<p>That should be available later today.</p>