

15-121, Spring 2018
Intro to Data Structures

Lecture #9 – Intro to Linked Lists
February 13, 2018

Mark Stehlik

Outline for Today

- HW2 (avg 87), Quiz3 (avg 88) questions?
- Some HW2, Q3 issues
- HW3 due next Monday
- Intro to Linked Lists

HW2 issues

- Always check array bounds! And close external files!
- Issues with removePerson?

```
for (int i = index; i < numContacts-1; i++)
    contacts[i] = contacts[i+1];
contacts[numContacts-1] = null; // why?
numContacts--;
```
- Using booleans as flags, not ints...
- What about a findPerson() helper function?
 - Return type? Visibility?
- No need for Person getters in Contacts toString()

Quiz 3 issues

- Memory diagrams?
- `public void setContact(Contact p, int index)`
 - needed if to make sure index was valid
 - which means?
 - `(index < numContacts && index >= 0)`

Linked Lists

- ArrayLists are quite useful since they remove the major issue with arrays – ArrayLists grow as you need them
- But, there are still issues (consider how much "wasted" space there is when the ArrayList grows) – there's no such thing as a free lunch in computer science
- Linked lists were the "original" answer to the compile-time bounded array issue
- Linked lists provide a programming foundation for working with other node-based data structures (i.e., trees & graphs)

Linked Lists

- Linked list is a collection of nodes
- Unlike an array/ArrayList, the nodes are not contiguous in memory
- Thus, to build a linked list of nodes, you need to know where the first node is, where the next node is...
- And, you need a way to mark the "end" (which will be?)

Linked Lists (definition)

- A linked list is
 - null (the only testable reference value)
 - a node whose next field stores (points to/refers to) a linked list
 - recursion, anyone?

List Node

- Needs to store (fields/attributes)
 - data (a String, for now)
 - the location of the next node (a reference to a Node)
- Methods
 - constructor
 - two getters (one for the data, one for the next)
 - one setter (the next)

Linked Lists

- `ListNode start = new ListNode("first", null);`
- Let's write the class and build a small test list
- As we work through the code, we'll be drawing lots of "box-and-pointer" diagrams. These will serve you incredibly well as a programmer; get used to drawing lots of them!