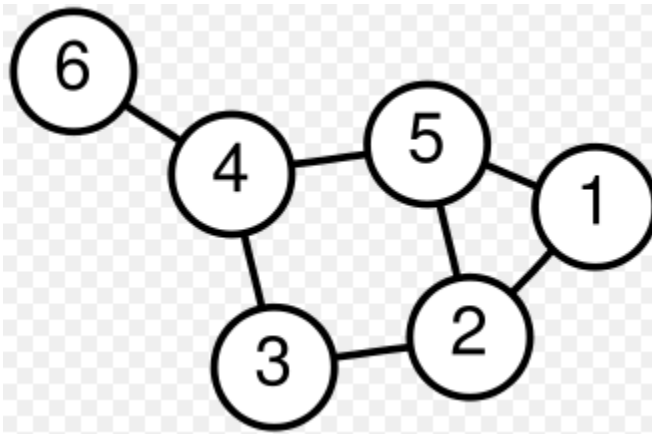


Recitation Guide June 13, 2007

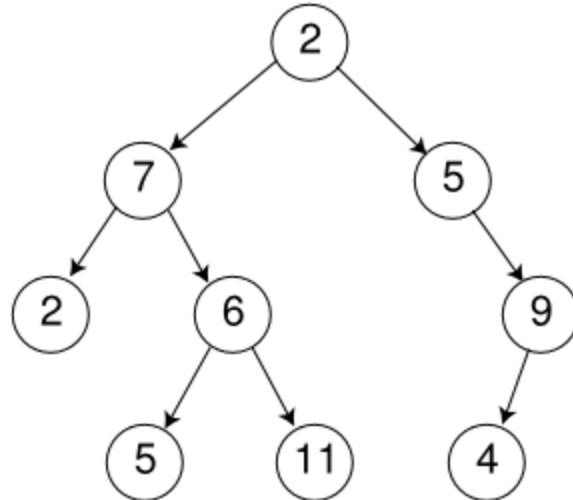
1. Test 1
 - a. Average 84(including zeros), 86 (excluding zeros)
 - b. Return Test 1 and go over solutions.
 - c. Remember that regrades will be handled by the head TA first and can then be appealed up to the professor.
2. Business Items
 - a. Homework 5 is due Monday, June 18. Grading Criteria already released to the coweb.
 - b. Quiz 2 is Wednesday, June 20.
 - i. Quiz grades will probably up before or on drop day.
 - c. Drop Day is Friday, June 22
3. Pair Programming
 - a. General introduction and overview – <http://coweb.cc.gatech.edu/cs1316/188>
 - b. Advise students to start looking for a partner
 - i. Pairs Page – <http://coweb.cc.gatech.edu/cs1316/704>
 - ii. Pair Request Page – <http://coweb.cc.gatech.edu/cs1316/705>
4. Linked Lists rehashed
 - a. PositionedSceneElement, SceneElementPositioned – Pictures placed one after the other.
 - b. LayeredSceneElement, SceneElementLayered – Pictures placed one on top of the other.
 - c. Introduction to inheritance with SceneElementPositioned and SceneElementLayered
 - i. Notice all of the repeated code in PositionedSceneElement and LayeredSceneElement. So instead we create a super class called SceneElement that contains all of the repeated code and have SceneElementPositioned and SceneElementLayered inherit from it.
5. Homework 5
 - a. Writing new linked list methods in PositionedSceneElement.
 - b. Homework 5 description – <http://coweb.cc.gatech.edu/cs1316/633#hw5>
 - c. Good methods to use
 - i. PositionedSceneElement
 1. remove(PositionedSceneElement node) – removes specified node from list and fixes all of the links
 2. getPicture() – returns the node's Picture
 3. insertAfter(PositionedSceneElement node) – inserts specified node after the node it is called on
 4. copy() – returns a copy of the node it is called on
 5. last() – returns the last node in the list
 6. drawFromMeOn(Picture bg) – draws the Picture from the node it is called on to the end of the list on a specified Picture.
 - ii. SimplePicture/ Picture
 1. getFileName() – returns the Picture's filename

- d. Looping through a linkedlist and writing a linkedlist method
 - i. Ask yourself
 1. Do I want to work with the current node or the one after it?
 2. What should my terminating condition be?
 3. What do I want to do at each node?
 4. Do I move to next node after I am done or stay in the same place?
 5. As there any special cases that might break my code?
 - a. Do I notify the user that this case has occurred? (Look at the remove method in PositionSceneElement. What happens when you try to remove the head of the list?)
 - e. Handling all cases/ possible user error
 - i. Remember that the student is the programmer in this course and he or she need to think about all of the possible inputs a user can put into the program.
 - ii. The more cases you consider the less likely your code will break due to user error.
 - iii. Notifying the user
 1. `System.out.println("debug");` //prints out debug and adds a carriage return so that subsequent lines will occur on the next line
 2. `System.out.print("debug");` //prints out debug and does not add a carriage return

6. Abstract data structures: Trees



- a. Graphs –main distinguishing characteristic is that they can have cycles.



- b. Trees –acyclic graphs or graphs that CANNOT have cycles
- i. Representing Trees in CS1316: LinkedLists of LinkedLists
 1. Breaking down the different branches
 - a. Picture Structures
 - i. Branch – can hold HBranch, VBranch, MoveBranch
 - ii. HBranch – can hold BlueScreenNodes; draws nodes horizontally
 - iii. VBranch – can hold BlueScreenNodes; draws BlueScreenNode vertically
 - iv. MoveBranch – can hold BlueScreenNodes; can move
 - v. BlueScreenNode – can hold Pictures.
 - b. Sound Structures
 - i. SoundBranch – can hold Sounds
 - c. There are more but these as the ones we tend to focus more on. Also it is also likely that these structures can hold other objects as well but we tend to focus on the ones mentioned above.
 - ii. FrameSequence: Formal introduction
 1. Will be used on homework 6
 2. Basically takes a snapshot of the current background Picture and saves it as a Picture.
 3. After you have taken all the snapshot you need, you can play them back like a movie.
 4. See example in WolfAttackMovie.java
 - iii. Seeing concepts in action: WolfAttackMovie.java
 1.

```
WolfAttackMovie wam = new WolfAttackMovie();
wam.setUp();
wam.renderAnimation();
wam.replay();
```

2. The version of WolfAttackMovie.java posted on the coweb has a main method that I added so you can just run it.
3. Do worry if FrameSequence is slow. It is supposed to be slow. For faster results try fewer frames and drawing on a white background.
4. Just so you know WolfAttackMovie is absurdly similar to homework 6.