3134 Data Structures in Java

Lecture 6
Feb 5 2007
Shlomo Hershkop
Announcements

- Hope you have been making progress on the homework
  - Hopefully you have already completed most of it

- Wednesday Feb 21 – Midterm
  - Open book
  - Will have a review session before hand

- Reading:
  - Chapter 3.1 – 3.4
Outline

- Lists
  - Runtimes
  - Code
- Iterators
Review

- What is an ADT ??

- What is the list ADT ??

- What are the runtimes on the list operations ??
  - Why?
Question

- Anyone know the difference between Mathematical induction vs Logical deduction?
Deduction

- Inference in which the conclusion about particulars follow from the general or universal premise

1. The picture is above the desk.
2. The desk is above the floor.
3. Therefore the picture is above the floor
Wrong deduction

1. Every terrorist opposes the government
2. Everyone in the opposition party opposes the government.
3. Therefore everyone in the opposition party is a terrorist

- what is wrong here?
Induction

- Inference of generalized conclusion from particular instances
  - i.e. the process of reasoning in which the premises of an argument support the conclusion but do not ensure it

- The Street is wet
- When it rains the street becomes wet
- It must have rained
difference

- deduction is logical necessity

- Usually will see something and induce something
  - which might be disproved later
  - this is not the case with mathematical induction
SAP Method

- here is a quick and dirty method for mathematical induction
  - We will be doing this when analyzing some of the algorithms later on this semester
  - Don’t be put off if it’s not straightforward, we will get plenty of classroom practice

- Show
- Assume
- Prove
Show

- Here we show the theorem holds in the simplest case (base)
Assume

- Assume the theorem holds for a general case
- called inductive hypothesis
- Example: assuming the hypothesis to be true for some specific integer $k$. 
Prove

Prove that the theorem holds for the next larger case
Next

- Does this make sense??

- Can you do the same for:

- Given a $2^n$ by $2^n$ checkerboard with any one square deleted, it is possible to cover this board with L-shaped pieces.
Example
what is the base case??
What is the assumption ??
How would the proof go?
$2^k$ by $2^k$ sections
Another Example

- We can prove that for the Fibonacci number series such that
  \[
  F_0 = 1, \ F_1 = 1, \ F_2 = 2, \ F_3 = 3, \ F_4 = 5, \ etc
  \]

- \( F_i \leq (5/3)^i \)

- Won’t do it now, but we can using the SAP method
Back to DS
Iterators

- some data structures have an idea of a position in the list
- We would like to abstract that away so that it makes it easier to work with the list

- We can use a helper also known as Iterators which allow you to iterate over a group of items
Issues

- How do we get an Iterator
- What operations would we like to support
- when required to use?
When do we need Iterators?

- List manipulations can be made safer and easier using Iterators

Example:
- inserts
- range inserts
- deletion
- range deletion
public interface Iterator<AnyType>
{
        boolean hasNext();
        AnyType next();
        void remove();
}
Code example

- Lets take a look at some actual code for the list and linked list stuff we discussed last time

- Want to show you how its done
  - Array list
  - Linked list
Applications

- Tons of applications for List DS
- Depending on how you are planning on using your list, that is the implementation you should be thinking about
Next Time

- Finish homework
- Reading:
  - chapter 3.1-3.2
  - 3.3 - 3.6