Lists and Arrays and Methods

2-17-2012

Opening Discussion

- Do you have any questions about the quiz?
- Minute essay comments:
  - Other collection types.
  - Gandalf vs. Dumbledore
  - Approaching problems.
  - Difference between recursion and loops.
  - Cutting a Double to a number of digits.
  - Do you need to improve your recursive methods?
  - Not knowing what to ask.
  - Infinite arrays or undefined length?

More

- TA? Kristen Lund
- Anti-virus recommendation.
- Array size limit?
- Filling an array with user input.
- Grading assignments.
- Heterogeneous arrays and lists.
- Default case for match.
- Cyber-attack of automated cars.
- Lab hours on weekends.

Recap Arrays and Lists

- Creation
  - Array(5,7,4)
  - List(8,5,3)
  - new Array[Double](1000)
  - 1::2::3::Nil
- Comparison
  - Arrays: mutable, fixed size.
  - Lists: immutable, :: to make new, longer list
- Indexing: start at 0
  - arr(5), arr(5)=”hi”

Using Lists

- You can do direct access on lists, but it is inefficient.
- The better method is to use the head and tail methods.
- The elements in a list can't be changed. However, you can efficiently add new elements at the front to make a new list.
- Lists work very well with recursion.

List and Array Patterns

- You can make patterns with Lists and Arrays.
- For Arrays:
  - Array(1,2,a,b,c)
- For Lists:
  - List(1,2,a,b,c)
  - h::t - matches any non-empty list
  - Nil - matches an empty list

Standard Methods

- There are lots of methods on collections. The API can help us see all of them.
- Part of collections:
  - drop, init, last, slice, splitAt, take, takeRight
- Boolean tests:
  - contains, endsWith, isEmpty, nonEmpty, startsWith
- Searching:
  - indexOf, lastIndexOf
- Other:
  - mkString, reverse, zip, zipWithIndex

Other Methods

- If the elements in a list support addition or multiplication, you can use the sum and product methods.
- If they are ordered you can do min and max.
- Having sum and length makes averages really easy.
- With min you can even drop a grade easily.

Higher Order Methods

- The most powerful methods are ones you can pass functions into.
  - exists, forall – Boolean checks like for math.
  - filter, partition – separate collection based on Boolean.
  - map – apply function to all the elements.
  - reduceLeft – apply function moving through collection
  - foldLeft – apply function moving through, but allows initial value so it can return a different type. This is curried.

Let's Put These Into Action

- I want to spend the rest of the class time playing with these methods and seeing what we can do with them.
- A String is a collection so you can do these things with a String as well.
- String also has a method called split.

Minute Essay

- What questions do you have? What collection method made the least sense?