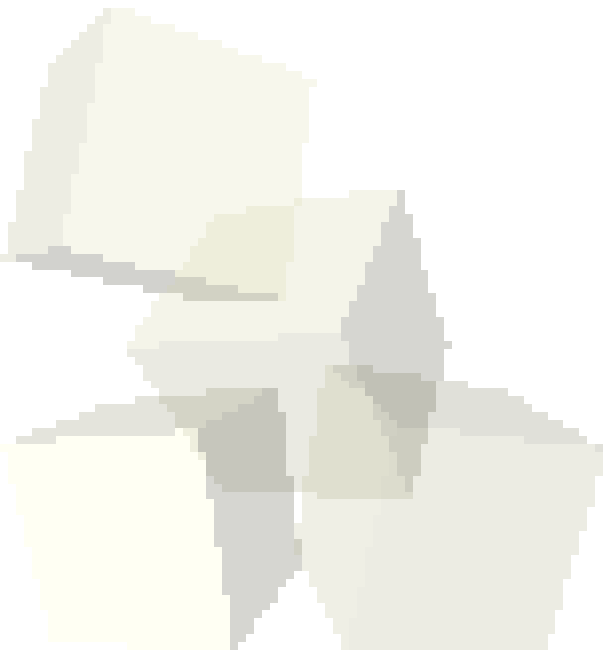




Sorting and Searching

2-9-2009





Opening Discussion

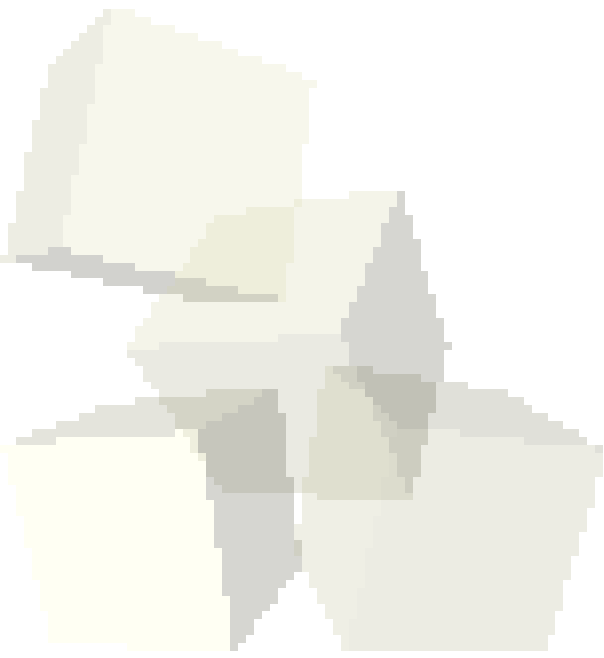
- Let's look at solutions to the interclass problem.
- Do you have any questions about the reading?
- Do you have any questions about the assignment?
 - ◆ Let's talk about how to submit the assignment.
- How to write `replaceAll`.
- Minute Essay Comments
 - ◆ Following along in class.
 - ◆ Word wrapping your code in Eclipse.
 - ◆ What is up with regular expressions?



- Array syntax in Java is just like it was in C as far as using the arrays. There are significant differences though.
- Array types are made by putting [] after a regular type.
- Array types are object types so your variables are references and should be instantiated with new.
- Arrays have bounds checking and know their length.
- You can't make arrays of generics.
- Each element in the array is like a variable. So for primitives they store values while for object values they are references.



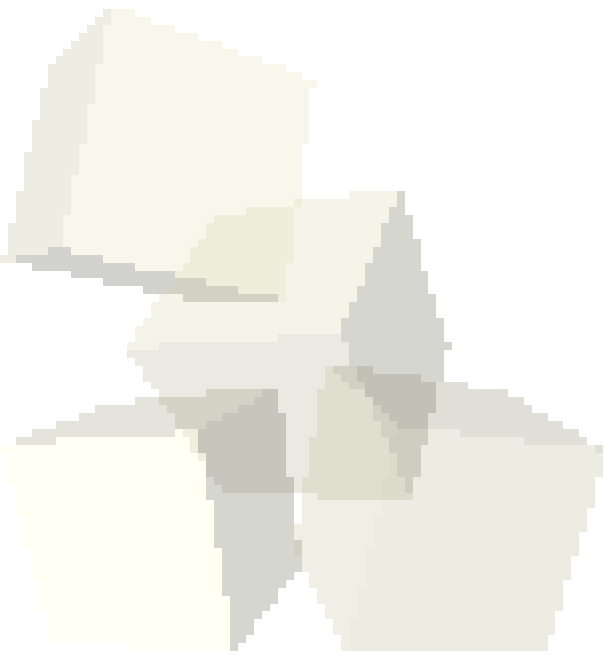
- I want to write a class called `ArrayMap` that we can use to use to make our `CommandProcessor` more flexible.
- It will also help to demonstrate use of arrays and generics.
- Let's work on that quickly.





Sorting and Searching Arrays

- These are topics that you should have talked about a fair bit in PAD1 so I'm not going to lecture on them much now.
- Instead, we'll write some code that uses arrays and these concepts.

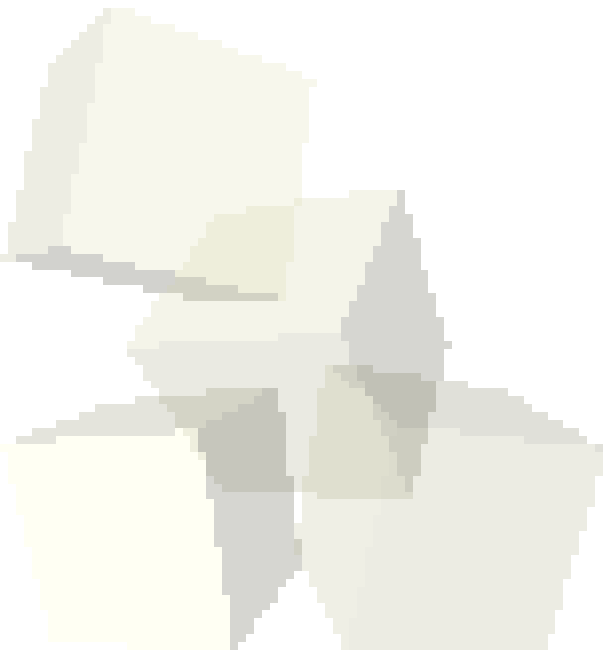




- A function $g(n)$ is $O(f(n))$ iff

$$\exists n, c : \forall m > n, c * f(m) > g(m)$$

- Let's look at what this means graphically.





- One of my motivating examples for polymorphism was a sort. In C you have to write a separate sort for every type, or you have to do some very odd stuff. In Java we can write polymorphic sorts of object types in at least two ways.
- You can write a sort/search that only takes subtypes of Comparable.
- You can write a sort/search that works on any Object, but that also takes an object of type Comparator.
- I prefer the second method as it is far more flexible.
- The `java.util.Arrays` class contains some utility methods.



- Let's write a method that uses one of the sorts you know to sort any object type. Try to make this a generic method so that it will be type safe. You can put it in a class called `ArrayHelper`.
- Let's make it so our comparator counts how many comparisons are made so we can see what sorts are best.
- This is something we can integrate into our command processing class. We can write a sort command that takes two arguments: sort type and number of elements.



- What sorts do you remember from PAD1? Explain how one of those sorts works.
- Remember to turn in your assignments by midnight and that quiz #2 is on Thursday.
- Interclass Problem – Write the best sort you know how to write in a polymorphic way using a comparator. Compare its performance to `Arrays.sort`.

