



Sorting and Searching

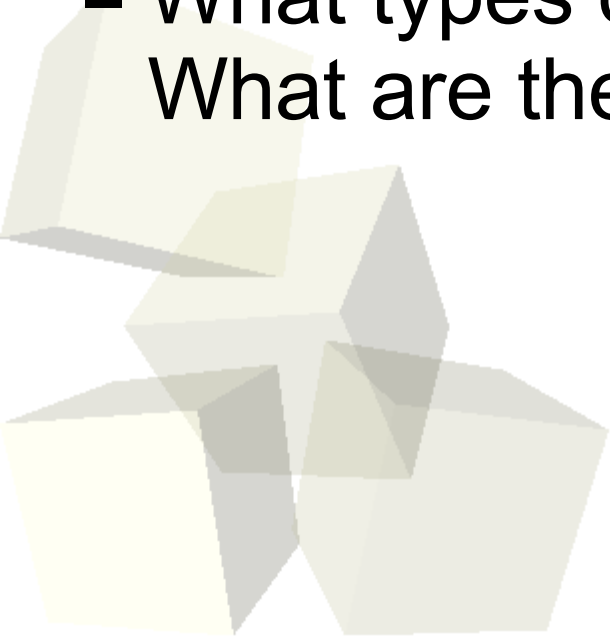
2/6/2007





Opening Discussion

- Do you have any questions about the reading?
- Do you have any questions about the assignment?
- What sorts did you learn in PAD1? How does each one work?
- What was the order of these sorts? What does that mean?
- What types of searches did you learn about? What are the orders of those?

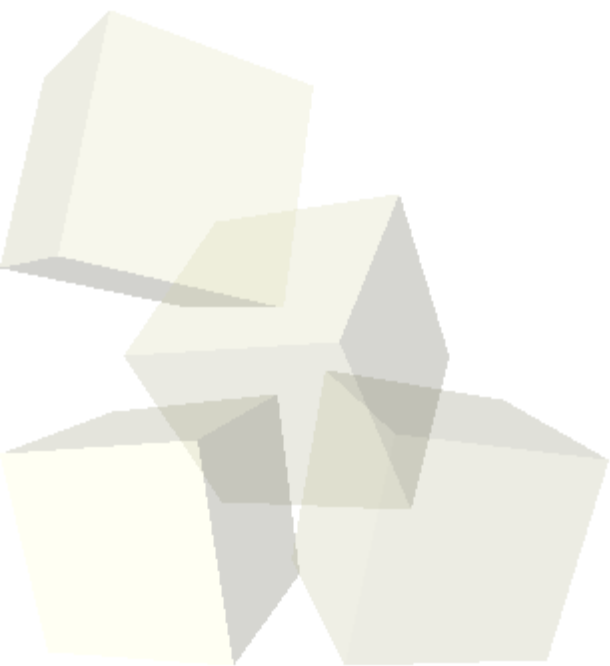




- A function $g(n)$ is $O(f(n))$ iff

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

- Let's look at what this means graphically.





Polymorphic Sorts

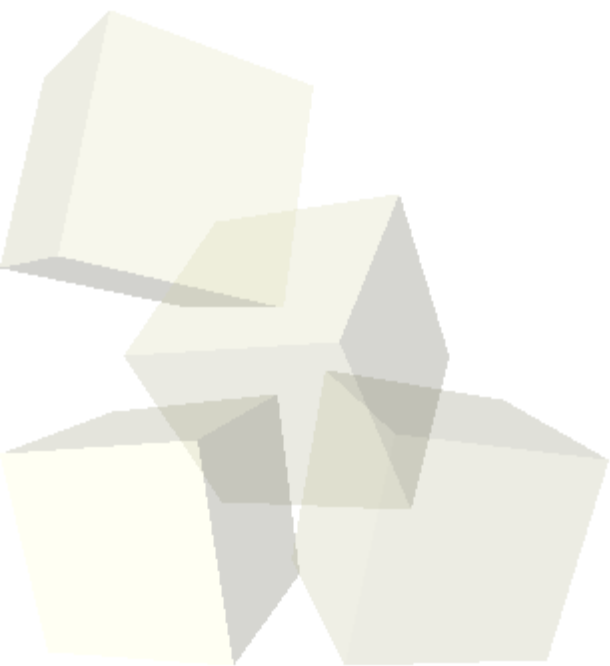
- One of my motivating examples for polymorphism was a sort. In C you have to write a separate sort for every type, or you have to do some very odd stuff. In Java we can write polymorphic sorts of object types in at least two ways.
- You can write a sort/search that only takes subtypes of Comparable.
- 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.



- I want you to 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 the *ArrayHelper* class you made last time.
- Let's make it so our comparator counts how many comparisons are made so we can see what sorts are best.
- If we have time we could write a search as well. Doing a non-recursive binary search would be nice because your book uses a recursive one.



- All the programming you have done so far has been in a single thread of execution. That is to say that the program goes from one line to the next doing one at a time in order. In a program with multiple threads the same thing happens, but in multiple places at once.





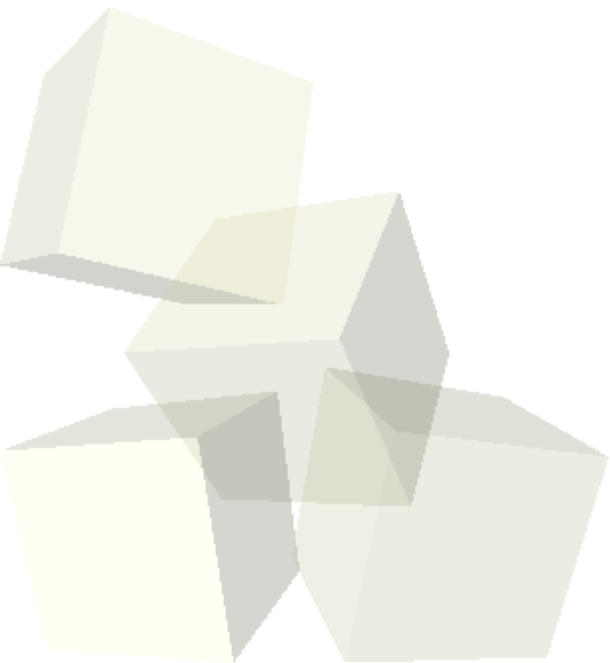
Why Threads?

- On a machine with a single processor and a single core threads simply give the impression of two things happening at once. With the widespread arrival of dual core processors, most of the new machines have the ability to actually do two things at once, assuming programs have more than one thread.
- For at least a while, the future is about adding more cores to processors so software is going to have to change and that means programmers have to change as well.



Threads in Java

- What are some of the key concepts associated with threads in Java? How long has Java been able to handle multiple threads?
- How do you create new threads?
- What are some of the issues associated with programs that have multiple threads? How do you get around these issues?





- Let's go ahead and write some code where we do some large sorts in parallel. Since threads are likely quite new to you we will do these as a class. You should also bring up the API and look at the Thread class.
- We can see how much of a speed up we can get on various machines in the department that have different numbers of processors.





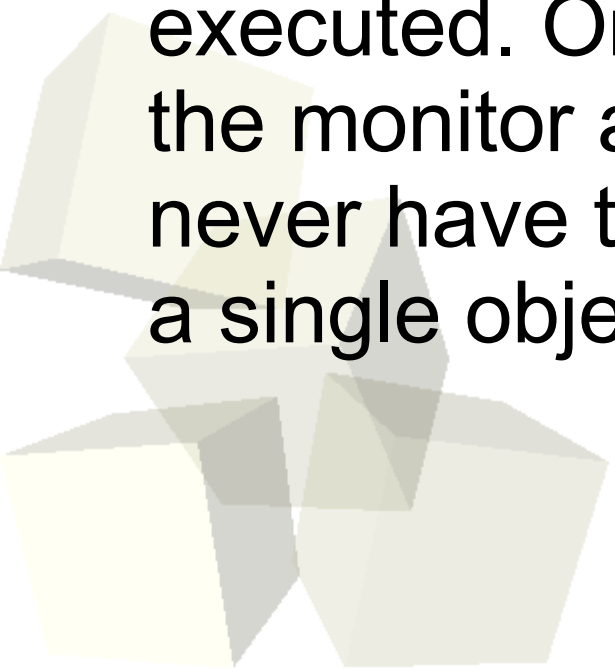
Thread Problems

- The primary problem one runs into with multithreaded programs is that threads share memory and more than one thread can access a piece of memory at once. This isn't a problem if they are just reading, but if any thread is writing you can have bad situations.
- An extreme condition would be to consider two threads operating on an array. Worst case is both are sorting the array at the same time. You could imagine one sorting while another tries to do a binary search and the results are similarly bad.
- The simplest (and most common) example is a bank account where a race condition occurs.



Synchronization

- The way to prevent two threads from accessing the same piece of memory at the same time is to synchronize the critical pieces of the code. You can put the synchronized keyword in front of methods or make synchronized blocks.
- Each object and class in Java can have a monitor that is locked when synchronized code is being executed. Only one thread can hold the lock on the monitor at a given time. This insures that you never have two threads executing critical code on a single object at the same time.





Wait and Notify

- We can get even more control over how threads behave with the wait and notify methods.
- The wait method will stop the execution of a thread until some other thread tells it to continue execution. The notify and notifyAll methods are how threads tell other threads that they are supposed to wake up.
- All of these must be called by a thread that holds the monitor to the object they are being invoked on. Typically that means that are called from inside synchronized code.



- When might you have to write your own code for doing a sort?
- The design for assignment #2 is due today. I will be in one of the labs (probably this one) from 2:00 until around 4:00.

