Homework 8

In this homework we will be doing some multithreading. In the homework zip file, in addition to these instructions there is a Command.java file that defines the Command interface, a QuickSort.java file that performs a quicksort, and a Main.java file that has the main function, which does several quicksorts. Your job is to modify this code to allow multiple things to happen at once in different threads.

What your program needs to do:

When a class implements the Command interface it promises to implement two functions: the function **public String identify( )** and the function **public void execute( )**. The identify function prints out some information about the object and the execute function does the work of the object.

There will be two classes that will implement the Command interface:

The first is the QuickSort class, which will be a modified form of the QuickSort I gave you. In this class the identify function will print out “quicksort of length “ followed by the length of the array and the toString() information for a quicksort object. The toString( ) method is provided. The execute function will initiate a quick sort of the QuickSort object’s arr element.

The second class is the DotProduct class. The DotProduct class has a constructor that takes a single int argument that defines the length of the arrays that will be multiplied together. The arrays will be initialized with random numbers. The identify function will print “inner product on arrays of length” followed by the array length, followed by “, the result is “ followed by the value of the result variable. The execute function will perform the dot product.

You will also implement a WorkQueue class. The WorkQueue class uses an ArrayList of Command references to hold work that is to be done. The method **void put(Command)** will insert a reference to a Command object (either a DotProduct object or a QuickSort object) into the queue. The method **Command get( )** will return a reference to a Command object from the queue, and remove the reference from the queue.

The main function will create a work queue, add 10 quicksorts and 10 dot product object references to it, create T threads and have them execute the work in the queue. The threads should be able to remove either a QuickSort or DotProduct reference from the queue and execute it and call identify on it after executing it. For the problem size, I started with 32000 and doubled it after creating each quicksort/dot product combination, i.e, I doubled it 10 times, total, not 20. If your program is running too slowly, and while doing initially debugging, feel free to reduce the problem size. Get a time immediately before creating the threads (use can use System.nanoTime( ) to do this) and print out the elapsed time after all threads have executed the routine.

Run your program with 2, 4 and 8 threads and observe the execution time.

What to turn in:

Turn in your code in a directory call *userid*. We should be able to go into the *userid* directory, execute *javac Main.java* followed by *java Main* and run your program.

Points will be given as followed:

2 points if your program runs and compiles

2 points for implementing the work queue and adding and removing references to objects to be executed from it

2 points for threads being able to execute any Command object

2 points for properly synchronizing your program

2 points for waiting for all threads to finish before taking the final time.