

CAS CS 112 – Spring 2011, Assignment 1

Problems 1-3 due at 10:00 pm on Thursday, February 3

Problem 1, Approximating Roots

(40 points) The Newton-Raphson method can be used to approximate the roots of a function (the roots of $f(x)$ are the values x where $f(x)$ is zero). We start by taking a guess x_0 , and then improving it by considering the point where the tangent to $f(x)$ at x_0 intersects the x-axis, let us call this point x_1 . x_1 will be closer to the root, and we can find x_2 by finding the intersection with the x-axis of the tangent to the curve at x_1 , and so on. It is easy to see that given x_n ,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

How is this useful for finding roots of real numbers? Consider that finding the k th root of a non-negative real number n is equivalent to finding the root of $f(x) = x^k - n$. Your job is to write a recursive function, which given k , n , x_0 and $iter$, computes an approximation of the k th root of n by performing $iter$ steps of Newton-Raphson starting at x_0 . Submit a single file called **NewtonRaphson.java** which has a main containing a loop asking the user for arguments, and calls a recursive function **float approximate (int k, float n, float x0, int iter)** that returns the approximation. The program should terminate when a negative k is input. You can read more about the Newton-Raphson method here: http://en.wikipedia.org/wiki/Newton's_method.

Problem 2, Wally Gone Wild

(30 points) Consider an organism that reproduces asexually, let us call him Wally (like the Red Sox mascot). It takes a Wally three generations to reach maturity, and a Wally never dies. Upon reaching maturity, a Wally gives birth to another Wally, and keeps giving birth to another Wally every generation thereafter. If in generation 1, we have one newborn Wally that matures and gives birth to another Wally in generation 4, how many Wallies will there be in generation n ? Please formulate a recursive and iterative solution to the problem. What is the complexity of each one (in terms of n)? Submit a single file called **Wally.java** with two methods: **countWallyRecursive** and **countWallyIterative**. Your main should be in the same file, it should repeatedly read an integer n from the console and print out the total number of Wallies in generation n twice (once using each function). The main loop should stop when a negative n is entered. State the complexity of each function in the comments. You can read about Wally and his rivalry with Raymond, the Tampa Bay mascot, here: http://en.wikipedia.org/wiki/Wally_the_Green_Monster.

Problem 3, Josephus problem

Problem 1.3.37 from Chapter 1 of our textbook on p. 168. Submit a single file called **Josephus.java** that implements your Queue client. Your client should call methods of the Queue API defined on p. 121. One implementation of the Queue ADT is given on p. 151.