CMPS 1600 Introduction to Computer Science II – Spring 14

4/11/14

10. Homework

Programming portion due **Thursday 4/17/14** at 11:55pm on Blackboard.

All the code for this homework should be in Scheme. Please submit one `.rkt` file on Blackboard.
In order to receive any credit for the programming portions, you are required to thoroughly comment and test your code.

1. **Merge Sort (11 points)**
   (a) (3 points) Write a function that returns a list of n random numbers. Use `(random)` to obtain a single random number.
   (b) (5 points) Implement the `merge` function that takes two sorted lists as parameters and returns the merged sorted list.
   (c) (3 points) Now use the code from the slides in class to implement mergesort. Test it with arrays of random numbers that are generated using your function above.

2. **Odd and Even (4 points)**
   All the recursive functions we have defined so far have been directly recursive, i.e., each function directly applied itself to a new argument. It is also possible to write two functions that use each other, resulting in indirect recursion. Write recursive functions `isOdd` and `isEven`, each in terms of the other. The functions should return the correct truth value for a non-negative input integer, depending on whether the integer is even or odd. Your functions should be recursive and not use the remainder function.

   Extra credit (1 point): Can you make your functions work for integers (negative or non-negative)?

3. **RemoveMultiple (5 points)**
   Write a function `removeMultiple` which takes a list `L` and a number `a` as arguments, and it returns a new list containing all the numbers in `L` that are not multiples of `a`. *(Hints: (1) It may be easier to create the output list with numbers in reverse order. That is fine as is; or you can use* `reverse` *afterwards. (2) You may want to use the* `remainder` *function.*

   Extra credit (up to 5 points): Implement the Sieve-of-Eratosthenes to compute a list of all prime numbers up to a given input number.