Problem Set #2

Starters

18. Write a method that takes in a string and returns true or false depending on whether the string is a palindrome. [2]
19. Write a method takes in two arrays and returns true if they are equal. [3] (And no using the == operator will not work!)
20. Write a method that takes in a number n (where $n \geq 1$), and returns the smallest number divisible by all the numbers between $1..n$. [5]

Algorithms & Data Structures

21. Write an implementation of the selection sort algorithm. [10]
22. What is the asymptotic time complexity of the selection sort algorithm? [3]
23. What is the asymptotic time complexity of inserting an item to the front of a linked list? [3]
24. What is the asymptotic time complexity of retrieving an item from a linked list? [3]
25. What is the asymptotic time complexity of retrieving an item from an array list? [3]
26. What is the asymptotic time complexity of retrieving the first item from a linked list? [3]
27. What is the asymptotic time complexity of searching for an item from an array list? [3]
28. What is the asymptotic time complexity of inserting an item into a heap? [3]
29. What is the asymptotic time complexity of finding the maximum item in an array list? [3]
30. What is the asymptotic time complexity of finding the maximum item in a max heap? [3]
31. What is the asymptotic time complexity of turning an array into a heap? [3]
32. Write an implementation of heap sort, using your implementation of *MaxHeap*. [10]
33. What is the asymptotic time complexity of heap sort? [3]
34. Write an implementation of insertion sort? [10]
35. What is the best case asymptotic time complexity of insertion sort? [3]
36. What is the average case asymptotic time complexity of insertion sort? [3]
37. Write an implementation of a Fixed Capacity Queue. [10]
38. Write an implementation of a Fixed Capacity Stack. [10]
39. Write an implementation of an unbalanced Binary Search Tree. [20]

Recursive Algorithms

40. Write a recursive *fibonacci* that takes in a parameter, n, and returns the *n*th Fibonacci number. [5]
41. Write a recursive method *factorial* that takes in a parameter n, and returns the product of all the numbers between 1..n [5]
42. Write a recursive method *summation* that takes in two parameters *min* and *max* and returns the sum of all the numbers between *min* and *max* [5]