CIS 22C
Final Exam - Review

Best way to prepare for the midterm exam:
Eat right, get enough sleep, and exercise☺
Make sure you arrive at the exam early!
Also, review class notes and the corresponding chapters in the textbook. Then practice class and book exercises and problems, and class examples without consulting the notes.

2hours, 100 points, open book, open notes. Final exams papers will be retained for a period of 90 days from the exam date. The final exam will be similar to the midterm exams, emphasizing the material learned after the second midterm but covering the rest of the quarter as well.

Exercises // see next pages:
1. (5 Points) Algorithm Efficiency // Page 302, 3 a-h, 4, 5, 6, 8, 10; Page 303, 1, 2, 3
2. (5 Points) Stacks and Queues ADT
3. (5 Points) Doubly and Multi-Linked Lists
4. (5 Points) Recursion, General Trees & Binary Trees
5. (5 Points) Binary Search Trees
6. (5 Points) Hash Tables
7. (8 Points) AVL Trees // Page 600: 1a, 1c
8. (8 Points) Heaps // Page 506: Q1-6; Page 512: Q7, 8, 9; Page 516: Q10
            // Page 518: Q11, 12; Page 522: 1 - 3, 7, 8; Page 523: 1, 3, 4
9. (8 Points) Graphs // Page 632: 1-4, (5 - optional); 10, 13, 14 (16, 17 - optional)
10. (8 Points) Sorting // Page 323: Q9, Page 331, Exercise 12, 8
11. (8 Points) Data Structure Design

Problem:
12. (30 points)
    Write a function or pseudo-code for problems dealing with hash tables, etc. such as:
    • Exercise 6 a, page 489
    • Save selected nodes from a Binary Tree into a queue (or a stack): (such as the leaves, or nodes within a given range
    • Write an InsertHash using linear probing or quadratic probing
    • Given the adjacency list of an undirected graph, write pseudo-code that will determine if the graph is connected or not.

===

1. Algorithm Efficiency: What is the BigO? Defend your answer.

```
algorithm  fun ( n )
    if( n EQUAL 1 )
        print( n )
    else
        fun ( n - 1 )
        print( n )
        fun ( n - 1 )
    end if
end fun
```

===

2. Stacks and Queues ADT

(A). Show the contents of **stack** and **que** after the following algorithm is executed.
(B). Explain why the below algorithm has a memory leak

Input sequence: **10, 10, 1, 2, 3, 10, 10, 10, 5, 6, 10**

```
loop( not end of input )
   allocate ( data and store its address in dataPtr )
   read( data using dataPtr )
   if( data not 10 )
       stack.push( dataPtr )
   else
       loop(not stack.isEmpty())
          stack.pop(dataPtr)
          que.enqueue( dataPtr )
       end loop
   end if
end loop
```

===

11. Data Structure Design (A question based on the Team Project).

You have been assigned to create a data structure design that uses a large list of books (up to 10000). The list of books needs to be searched by ISBN. Also, it needs to be displayed sorted by title and/or sorted by author. In addition, the list needs to be searched by title and by author.

Draw a data structure for this application and briefly explain why you choose it. List any assumptions used in the design.