CS210a Data Structures and Algorithms
Assignment 3
Due date: October 26
Total of 20 Marks

Put your assignment in a 9" × 12" envelope labelled with your name, course number and CS220 class list number and drop it in the CS2210 locker (locker #300 located on the third floor of the Middlesex College Building) by 11:59 pm on the due date. You must also include a signed submission form.

You might find these facts useful: \( \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}, \sum_{i=0}^{n-1} (n - i) = \sum_{i=1}^{n} i. \)

1. (2 marks) Consider a hash table of size \( N = 7 \) where we are going to store integer values. The hash function is \( h(k) = k \mod 7 \). Draw the table that results after inserting, in the given order, the following values: 15, 5, 33, 12, 47. Assume that collisions are handled by separate chaining.

2. (2 marks) Show the result of the previous exercise, assuming collisions are handled by linear probing.

3. (2 marks) Repeat exercise (1) assuming collisions are handled by double hashing, using secondary hash function \( h'(k) = 5 - (k \mod 5) \).

4. (3.5 marks) Solve the following recurrence equation and give the order of \( f(n) \). You must explain how you solved the equation.

\[
\begin{align*}
    f(1) &= 1 \\
    f(n) &= f(n-1) + n - 2
\end{align*}
\]

5. (i) (7 marks) A tree is called symmetric if for every internal node \( u \) all the children of \( u \) store the same value. For example the following tree with root \( a \) is symmetric, but the tree with root \( b \) is not symmetric as one child of node \( v \) stores value 2 while the other stores 3.

Write in pseudocode an algorithm \( isSymmetric(r) \) that receives as input the root \( r \) of a tree and it outputs the value \( true \) if the tree is symmetric and \( false \) if it is not. For a node \( v \) let \( v.value \) denote the value stored in \( v \).

5. (ii) (3.5 marks) Compute the worst case time complexity of your algorithm as a function of the total number \( n \) of nodes in the tree. You must

- explain what the worst case for the algorithm is
- explain how you computed the time complexity
- give the order of the time complexity of the algorithm