SCS 1308 – FoA

Q1. Solve the Following Recurrence Relations using the Recursion Tree Method

a) $T(n) = 2T(n/2) + 1$
b) $T(n) = T(n/2) + n$

Q2. Solve the Following Recurrence Relations using the Iteration Method

a little too hard

a) $T(n) = 2T(n/4) + n^2$
b) $T(n) = 3T(n/3) + n$

Q3. Solve the Following Recurrence Relations using Master Theorem

$$T(n) = a\,T(n/b) + f(n)$$
$$T(1) = c$$

where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$ then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

a) $T(n) = 3T(n/2) + n^4$
b) $T(n) = 7T(n/2) + n^2$

Q4. Prove that the sum of the first n odd numbers is n^2

Q5. Prove that the sum of the first n even numbers is n(n+1)

Q6. Prove the correctness of the following code using Loop Invariant

```
int func(int arr[], int size) {
    int sum = 0;
    for (int i = 0; i < size; ++i) {
        if (arr[i] % 2 == 0) {
            sum += arr[i] * arr[i];
        }
    }
    return sum;
}
```

Q1. Compare and Contrast Binary Search and Interpolation Search

Q2. Compare and Contrast Linear Search and Jump Search

Q3. Perform Shell Sort on the following list of numbers. Show all steps

> **not really thought through, go with gap sizes 8 – 4 – 2, and then insertion sort**
>
> 34, 65, 23, 12, 66, 28, 44, 11, 7, 76, 34, 98, 63, 87, 17, 31

Q4. Perform Radix Sort on the following list of numbers. Show all steps

> 845, 1723, 34, 405, 189, 506, 91, 1132, 204, 6980, 55, 611, 299

Q5. Insert the following values in order to a hash table with the corresponding hash function

a) 65, 23, 76, 29, 68, 21, 98, 34, 12, 56
h(k) = k % tableSize
tableSize = 7
Linear Probing for Collisions

b) 89, 34, 9, 11, 64, 21, 76, 43
h(k) = k % tableSize
tableSize = 11
Quadratic Probing for Collisions
$h'(k, i) = (h(k) + i^2) \% \text{tableSize}$

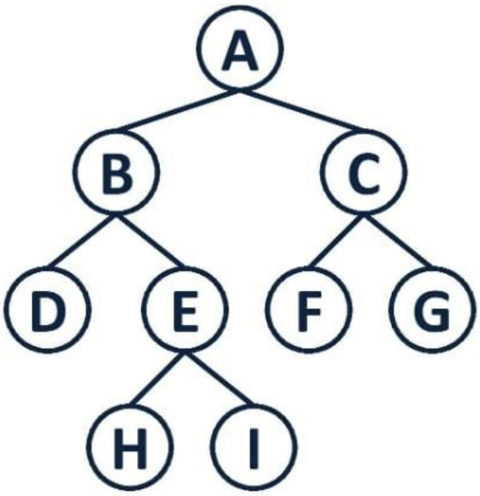
c) 12, 66, 28, 44, 98, 63, 87
$h_1 = (k + 5) \% \text{tableSize}$
tableSize = 17
Double Hashing for Collisions
$h_2 = (k + 1) \% \text{tableSize}$
$h'(k, i) = (h_1(k) + i \times h_2(k)) \% \text{tableSize}$

Q6. What is Rehashing. When and Why is it done

Q1. Define the Following

a) Simple Graph
b) Strongly Connected Graph
c) Bipartite Graph
d) Forest
e) Complete Binary Tree
f) Full Binary Tree
g) Perfect Binary Tree
h) Binary Search Tree

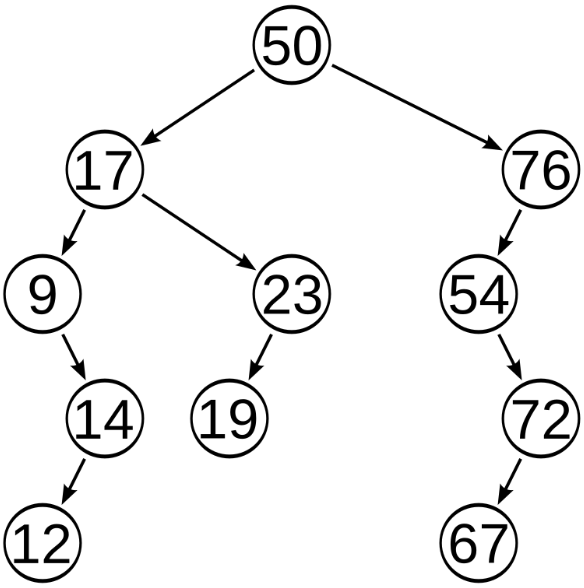
Q2. What will be the In-Order, Pre-Order, Post-Order and Level-Order Traversals of the following Binary Tree



Q3. Write Pseudocode for the following scenarios

a) Searching for an element in a Binary Search Tree
b) Finding the maximum element of a Binary Search Tree
c) Printing elements in a Binary Tree in Post-Order Traversal
d) Printing elements in a Binary Search Tree in Level-Order Traversal
e) Finding the height of a Binary Tree

Q4. Balance the following Binary Search Tree using the DSW Algorithm. Show all steps



Q5. Explain what the balance factor is in an AVL tree, and how it is used to maintain a balanced tree

Q6. Perform the following insertions (in order) on an empty AVL tree

21, 26, 30, 9, 4, 14, 28, 18, 15, 10, 2, 3, 7

Q7. On the final tree generated from the previous question, delete the following elements in the specified order

18, 15, 4

Q8. What are the 4 properties of a Red-Black tree

Q9. Explain the steps taken when inserting an element into a Red-Black tree.

Q9. What are the 4 properties of a Red-Black tree *(Specify details on all cases of Red-Black violations and how they are addressed)*

Q10. Perform the following insertions (in order) on an empty Red-Black tree

1, 10, 5, 7, 3, 13, 6, 4, 8, 9, 7, 0