1. Know the different array-based and reference-based implementation of a binary tree ADT.
2. Know the three tree traversals: inorder, postorder and preorder.
3. Understand the Binary Search Tree (BST) properties and know how to perform an insertion, deletion and search.
   (A cool visualization where you can practice: https://www.cs.usfca.edu/~galles/visualization/BST.html)
4. Understand the concepts of Full, Balanced and Complete binary trees.
5. Know what is a root, leaf, ancestor, sibling nodes and the height of the tree.
6. Know the following theorem from chapter 11:
   Theorem 11-2
   A full binary tree of height \( h \geq 0 \) has \( 2^{h-1} \) nodes
   Theorem 11-3
   The maximum number of nodes that a binary tree of height \( h \) can have is \( 2^h - 1 \)
7. Know hashing and all the collision resolution strategies (linear probing quadratic probing double hashing separate chaining rehashing).
8. Know the approximations of the number of probes for every collision resolution strategy.
9. Understand the Big-Oh notation.
10. Read carefully all the slides covered in class and ask me on Monday in case of any doubt !!!!
Midterm Exam II: Review Problems

PART I: Problems:

Problem 1:
(a) Draw the binary search tree that is created if the following numbers are inserted in the tree in the given order.
12 15 3 35 21 42 14 12

(b) What is the height of the tree in (a)? 4
(c) Draw a balanced binary search tree containing the same numbers given in part (a).
Many solutions for this question. Example of balanced tree

(d) What is the height of the tree in (c)? 3
(e) Store the resulting tree in (c) using the array-based implementation of the binary tree

| 15 | 12 | 35 | 3 | 14 | 24 | 42 | 21 |

Problem 2:
(a) Draw the binary search tree that is created if the following numbers are inserted in the tree in the given order.
Problem 3: Consider the following BST:

(b) Delete node 5 from the tree and draw the resulting tree
(a) What is the sequence of visited nodes if the tree is traversed using inorder traversal?
6, 40, 8, 9, 20, 2, 97, 76
(b) What is the sequence of visited nodes if the tree is traversed using postorder traversal?
6, 8, 9, 40, 2, 76, 97, 20
(c) What is the sequence of visited nodes if the tree is traversed using preorder traversal?
20, 40, 6, 9, 8, 97, 2, 76

Problem 4:
Consider the following Binary tree. This is an incorrect BST. To make it proper, exactly one swap must occur.

(a) Circle the two numbers that needs to be swapped.

(b) What is the maximum number of operations needed to search for any node in the tree? 4 (search 40)

Problem 5: Are the following statement True or False?

3000N + 36 = O(N) → True
N log N + N^2 = O(N^2) → True
N log N + N^2 = O(N log N) → True
N^2 log N + N^2 = O(N^2) → True
N^{1/2} + log N = O(log N) → True

Problem 6: Answer the following questions
(a) How many nodes does a full binary tree of height 101 have?  
\[2^h - 1 = 2^{101} - 1\]
(b) What is the number of nodes in a full tree in level 5?  
\[2^{h-1} = 2^4 = 16\]
(c) Are all Full trees Complete? True False
(d) Are all Complete trees Balanced? True False
(e) Are all Full trees Balanced? True False
(f) Are all Complete trees Full? True False
(g) Are all Complete trees Balanced? True False

Problem 7:
You have a hashtable of size m=11 and a (not very good) hash function h:

\[h(x) = (\text{sum of the values of the first and last letters of } x) \mod m\]

where the value of a letter is its position in the alphabet (e.g., value(a)=1, value(b)=2, etc.). Here are some precomputed hash values:

<table>
<thead>
<tr>
<th>word</th>
<th>ape</th>
<th>bat</th>
<th>bird</th>
<th>carp</th>
<th>dog</th>
<th>hare</th>
<th>ibex</th>
<th>mud</th>
<th>koala</th>
<th>stork</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

(a) Draw a picture of the resulting hashtable (using chaining) after inserting, in order, the following words: ibex, hare, ape, bat, koala, mud, dog, carp, stork.

(b) Which cells are looked at when trying to find bird. 
Array at index 6

Problem 8:
Suppose you are given the following information about a hashtable.
Space Available (in words) 2000
Number of Items 1000
Proportion Successful Searches 1/3

(a) What is the expected number of probes for a successful and unsuccessful search operations when hashing with **separate chaining** is used? The load factor is $\alpha = 2000/1000=0.5$

- successful $1 + (\alpha/2) = 1.25$
- unsuccessful $\alpha = 0.5$

(b) What is the expected number of probes for a successful and unsuccessful search operations when hashing with **linear probing** is used?

- successful $0.5 \times [1 + 1/(1 - \alpha)] = 0.5 \times [1 + 1/(1 - 0.5)] = 1.5$
- unsuccessful $0.5 \times [1 + 1/(1 - \alpha)^2] = 2.5$

(c) What is the expected number of probes for a successful and unsuccessful search operations when hashing with **quadratic probing** is used?

- successful $-\log_e(1 - \alpha)/\alpha = -\log_e(1 - 0.5)/0.5 = 1.3$
- unsuccessful $1/(1 - \alpha) = 1/0.5 = 2$

(d) What is the expected number of probes for a successful and unsuccessful search operations when hashing with **double hashing** is used?

- successful $-\log_e(1 - \alpha)/\alpha = -\log_e(1 - 0.5)/0.5 = 1.3$
- unsuccessful $1/(1 - \alpha) = 1/0.5 = 2$

**Problem 9:** Draw a binary tree that contain the letters “b”, “n”, “o”, “s”, “u” such that the inorder traversal spells “bonus” and the preorder traversal spells “obuns”.

```
    o
   /|
  b n
 /|
u s
```

**Problem 10:**

Draw the contents of the hash table in the boxes below given the following conditions: The size of the hash table is 12. Double hashing is used to resolve collisions.

The hash function used is: $h_1(k) = k \mod 12$

The second hash function is: $h_2(k) = 7 - (k \mod 7)$

Double hashing function is: $h_3(x) = (h_1(x) + i*h_2(x)) \mod 12$

What values will be in the hash table after the following sequence of insertions? Draw the values in the boxes below.

33, 10, 9, 13, 12, 45, 26, 17
Collision 1: insert 9 → \( h_1(9) = 9 \), \( h_2(9) = 5 \), \( h_3(9) = 2 \)

Collision 2: insert 45 → \( h_1(45) = 9 \), \( h_2(45) = 4 \), \( h_3(45) = 1 \)
\[ h_3(45) = (9+2*4) \mod 12 = 5 \]

Collision 3: insert 26 → \( h_1(26) = 10 \), \( h_2(26) = 2 \), \( h_3(26) = 0 \)
\[ h_3(26) = (10+2*2) \mod 12 = 2 \]
\[ h_3(26) = (10+3*2) \mod 12 = 4 \]

Collision 4: insert 17 → \( h_1(17) = 5 \), \( h_2(17) = 4 \), \( h_3(17) = 9 \)
\[ h_3(17) = (5+2*4) \mod 12 = 1 \]
\[ h_3(17) = (5+3*4) \mod 12 = 5 \]
\[ h_3(17) = (5+4*4) \mod 12 = 9 \ldots . \]

CANNOT INSERT 17 !!! (12 is not a good table size)

Problem 11:

Hashing: Draw the contents of the hash table in the boxes below given the following conditions: The size of the hash table is 9. Quadratic probing is used to resolve collisions.

The hash function used is \( h(k) = k \mod 9 \)

(a) What values will be in the hash table after the following sequence of insertions? Draw the values in the boxes below. 10, 35, 18, 19, 26

\begin{align*}
10 \mod 9 &= 1 \\
35 \mod 9 &= 8 \\
18 \mod 9 &= 0 \\
19 \mod 9 &= 1 \\
&+1 \rightarrow 2 \\
26 \mod 9 &= 8 \\
&+1 \rightarrow 9 \mod 7 = 0 \\
&+4 \rightarrow 12 \mod 9 = 3
\end{align*}

(b) What is the load factor for the table above? \( \frac{5}{9} \)
(c) Is the value 7 a good choice for table size? Give one good reason for picking 7 and one bad reason for picking 7 for table size.

7 is a good choice because: it is a prime number

7 is a bad choice because: It is too small compared to the number of elements inserted. It will result in a higher load factor.

Problem 12:

Draw the contents of the hash table in the boxes below given the following conditions: The size of the hash table is 7. Linear probing is used to resolve collisions.

The hash function used is $H(k) = k \mod 7$

What values will be in the hash table after the following sequence of insertions? 8, 33, 15, 26, 22

Problem 13:

For each of the functions $f(N)$ given below, indicate the tightest bound possible.
Problem 14:
Describe the worst case running time of the following pseudocode functions in Big-Oh notation in terms of the n.

```java
void silly(int n) {
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < i; ++j) {
            System.out.println("j = " + j);
            for (int k = 0; k < j; ++k)
                System.out.println("k = " + k);
        }
    }
}
```

\[ O(n^3) \]
Problem 15:
What is the minimum and maximum number of nodes in a full binary tree of height h (use h in your answer)?

Minimum = $2^h - 1$

Maximum = $2^h - 1$

Problem 16: Given the following six trees a through f, List the letters of all of the trees that have the following properties:
Full: ________ C E ___________

Complete: ____ C E B D ___________

Balanced: _____ A B C D E F ___________

Problem 18:

Build the following tree using the Binary Tree API we have seen in class/lab:

```java
/* Binary Tree API
BinaryTree tree = new BinaryTree('b');
tree.attachLeft('a');
tree.attachRight('c');
tree1.attachRightSubtree(tree);
tree1.attachLeftSubtree(tree)
*/

BinaryTree<Character> tree1 = new BinaryTree("F");
BinaryTree<Character> rightsubtree1 = new BinaryTree("G");

BinaryTree<Character> leftsubtree1 = new BinaryTree("B");
leftsubtree1.attachRight('A');

BinaryTree<Character> leftsubtree2 = new BinaryTree("D");
leftsubtree2.attachRight('E');
leftsubtree2.attachLeft('C');

BinaryTree<Character> rightsubtree2 = new BinaryTree("I");
leftsubtree2.attachLeft('H');

leftsubtree1.attachRightSubtree(leftsubtree2);
rightsubtree1.attachRightSubtree(rightsubtree2);

tree1.attachLeftSubtree(leftsubtree1);
tree1.attachRightSubtree(rightsubtree1);
```

Problem 19:
Let $H$ be a hash-table where collisions are handled using rehashing (doubling the size of the hash table when the load factor is exceeded). Let’s assume that the load factor assumed for this problem is $0.5$. After inserting 10 elements with different keys what is the size of the hash table? 

**Problem 20:** Give the contents of the hash table that results when you insert items with the keys D E M O C R A T in that order into an initially empty table of $M = 5$ lists, using separate chaining. Use the hash function:

$$h(x) = (11^*k) \mod M$$

**e.g:** $h(I) = h(9) = (11*9) \; \% \; 5 = 4$.

```
<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>O</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>
```

**PART II: Multiple Choice Questions:**

1. The height of a a full binary tree of $n$ nodes is?
   a) $\log(n)$
   b) $\log(n+1)$
   c) $n^{1/2}$
   d) $\frac{n}{2}$

2. Which of the following is not an advantage of trees?
   a) Hierarchical structure
   b) Faster search
   c) Undo/Redo operations in a notepad

3. The number of nodes from the root node to the deepest leaf is called __________ of the tree.
   a) Height
   b) Width
   c) Length
   d) None of the mentioned

4. What is a full binary tree?
   a) Each node has exactly zero or one children
   b) Each node has exactly two children
c) All the leaves are at the same level  
d) Each node has exactly one or two children

5. What is a complete binary tree?  
a) Each node has exactly zero or two children  
b) A Full binary tree except the bottom level, which is filled from right to left  
c) A Full binary tree except the bottom level, which is filled from left to right  
d) None of the mentioned

6. What is the time complexity for finding an element in the binary tree in the worst case scenario?  
a) \( O(\log \log n) \)  
b) \( O(n \log n) \)  
c) \( O(n) \)  
d) \( O(\log n) \)

Suppose \( T \) is a binary tree with 14 nodes. What is the minimum possible height of \( T \)?  
a) 0  
b) 3  
c) 4  
d) 5

7. What are the children for node ‘w’ of a full-binary tree in an array representation?  
a) \( 2w \) and \( 2w+1 \)  
b) \( 2+w \) and \( 2-w \)  
c) \( w+1/2 \) and \( w/2 \)  
d) \( w-1/2 \) and \( w+1/2 \)

8. Consider a situation of writing a binary tree into a file with memory storage efficiency in mind, is array representation of tree is good?  
a) yes because we are overcoming the need of pointers and so space efficiency  
b) yes because array values are indexable  
c) No it is not efficient in case of sparse trees and remaining cases it is fine  
d) None of the above

9. Disadvantage of using array representation for binary trees is?  
a) difficulty in knowing children nodes of a node  
b) difficult in finding the parent of a node  
c) have to know the maximum number of nodes possible before creation of trees  
d) difficult to implement

10. Binary trees can have how many children?  
a) 2  
b) any number of children  
c) 0 or 1 or 2  
d) 0 or 1
11. How to travel a tree in linked list representation?
   a) using inorder traversing
   b) using preorder traversing
   c) using postorder traversing
   d) all of the mentioned

12. Disadvantages of linked list representation of binary trees over arrays?
   a) Randomly accessing is not possible
   b) Extra memory for a pointer is needed with every element in the list
   c) Difficulty in deletion
   d) Random access is not possible and extra memory with every element

13. What is the speciality about the preorder traversal of a binary search tree?
   a) It traverses in a non increasing order
   b) It traverses the left then right then root nodes
   c) It traverses the root left then right
   d) None of the mentioned

14. What is the speciality about the postorder traversal of a binary search tree?
   a) It traverses in a non increasing order
   b) It traverses the left then right then root nodes
   c) It traverses the root left then right
   d) None of the mentioned

15. Advantages of linked list representation of binary trees over arrays?
   a) dynamic size
   b) ease of insertion/deletion
   c) ease in randomly accessing a node
   d) both dynamic size and ease in insertion/deletion

16. Which of the following is false about a binary search tree?
   a) The left child is always lesser than its parent
   b) The right child is always greater than its parent
   c) The left and right sub-trees should also be binary search trees
   d) None of the mentioned

17. What is the specialty about the inorder traversal of a binary search tree?
   a) It traverses in a non increasing order
   b) It traverses in an increasing order
   c) It traverses in a random fashion
   d) None of the mentioned

18. How will you find the minimum element in a binary search tree?
   a) 
   
   ```java
   public void min(Tree root)
   ```
```java
{     
    while (root.left != null) 
    { 
        root = root.left; 
    } 
    System.out.println(root.data); 
}

b)
public void min(Tree root) 
{     
    while (root != null) 
    { 
        root = root.left; 
    } 
    System.out.println(root.data); 
}

c)
public void min(Tree root) 
{     
    while (root.right != null) 
    { 
        root = root.right; 
    } 
    System.out.println(root.data); 
}

d)
public void min(Tree root) 
{     
    while (root != null) 
    { 
        root = root.right; 
    } 
    System.out.println(root.data); 
}
```

19. What are the worst case and average case complexities of a binary search tree?
   a) O(n), O(n)
   b) O(logn), O(logn)
   c) O(logn), O(n)
   d) O(n), O(logn)

20. If several elements are competing for the same bucket in the hash table, what is it called?
   a) Diffusion
   b) Replication
   c) Collision
   d) None of the mentioned
21. In simple chaining, what data structure is appropriate?
   a) Singly linked list
   b) Doubly linked list
   c) Circular linked list
   d) Binary trees

22. What is a hash table?
   a) A structure that maps values to keys
   b) A structure that maps keys to values
   c) A structure used for storage
   d) A structure used to implement stack and queue

23. What can be the techniques to avoid collision?
   a) Linear Probing
   b) Quadratic probing
   c) Double hashing
   d) All of the mentioned

24. What is a hash function?
   a) A function has allocated memory to keys
   b) A function that computes the location of the key in the array
   c) A function that creates an array
   d) None of the mentioned

25. Which of the following statements about binary trees is NOT true?
   a) Every binary tree has at least one node.
   b) Every non-empty tree has exactly one root node.
   c) Every node has at most two children.
   d) Every non-root node has exactly one parent.

26. Is this a binary search tree?
   a) True
   b) False

27. Consider the binary tree below. Answer the following questions
(a) Which statement is correct?
   a) The tree is neither complete nor full.
   b) The tree is complete but not full.
   c) The tree is full but not complete.
   d) The tree is both full and complete.

(b) How many of the nodes have at least one sibling?
   a) 5
   b) 6
   c) 7
   d) 8
   e) 9

(c) What is the height of the tree?
   a) 3
   b) 4
   c) 8
   d) 9

28. Which of the following functions grows fastest?
   (a) n log n
   (b) \(2^n\)
   (c) log n
   (d) \(n^2\)
   (e) \(n^{200}\)

29. Which of the following functions grows fastest?
   (a) n + log n
   (b) n log (2n)
   (c) n log n
   (d) n
   (e) There is a tie among two or more functions for fastest growth rate