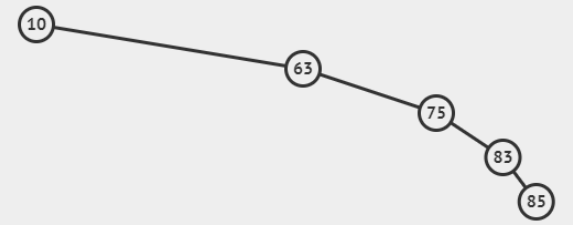
CS 261-020 Exam II Winter 2022

Part I: True (T) / False (F), put T/F on the answer sheet (32 pts, 2 pts each)

1. If all nodes in a structure have at most one parent, then this structure can be counted as a tree.

2. A binary tree that every interior node has exactly two children is called a complete binary tree.

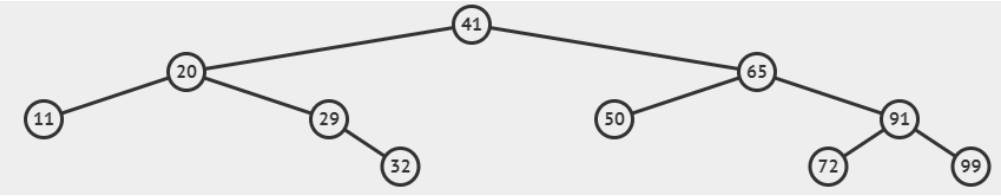
3. The following is a binary search tree.



4. In BST, the in-order successor of a node N is always the leftmost node in N's right subtree.

5. In an unbalanced BST, the runtime complexity for removing a node is always O(log n) where n represents the number of nodes in the tree.

6. The following BST is balanced.



7. In an AVL tree, the runtime complexity for inserting a node is always O(log n) where n represents the number of nodes in the tree.

8. The **`first()`** function returns and removes the first element in the priority queue.

9. Given a heap, we can perform an O(log n) heapsort without allocating additional memory.

10. It is efficient to implement an incomplete binary tree using an array.

11. The data elements in a map are composed of key-value pairs.

12. A hash table tends to perform better overall (both time and space) than an array, linked list, or balanced BST.

13. When designing a hash function, we should make its algorithm as complex as possible so the inputs can be mapped as evenly as possible over the output range.

14. A perfect hash function is one that results in no collisions for a table size that equals the number of elements.

15. When representing a graph, the space complexity of an adjacency matrix is O(|V| + |E|), where V and E represents the number of vertices and edges, respectively, in the graph.

16. Linear probing is not a good solution for clustering.

Part II: Multiple Choices. Put your answers on the answer sheet (40 pts, 2 pts each)

1. Which of the following is not an application of tree?
   A. Computer's filesystem
   B. Compiler's abstract syntax of a program
   C. Object model of a web page
   D. The "back" operation in web browsers

2. The ancestor of all other nodes in the tree is called _____.
   A. root
   B. parent
   C. height
   D. child

3. The _____ of a node are all of its children, plus its children's children.
   A. subtrees
   B. descendants
   C. siblings
   D. leaf nodes

4. If a perfect binary tree has height h, then how many nodes does it have?
   A. 2^h
   B. 2^{h+1}
   C. $2^h - 1$
   D. $2^{h+1} - 1$

5. Which of the following is not a depth-first tree traversal?
   A. In-order
   B. Pre-order
   C. Post-order
   D. Level-order

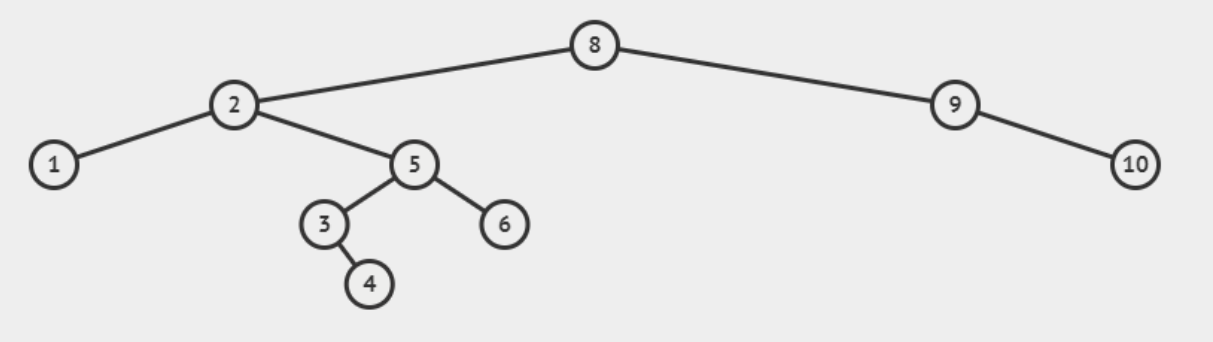
6. A complete binary tree where every node's value is less than or equal to the values of its children is called?
   A. Hash table
   B. Map
   C. Minimizing Heap
   D. Priority queue

7. An operation that moves nodes up/down the tree according to their priority values is called?
   A. Traversal
   B. Searching
   C. Percolation
   D. Hashing

8. Suppose a hash table has 5 elements in it. After attempting to insert 5 duplicate keys into the same hash table, what is the new size?
   A. 0
   B. 5
   C. 10
   D. 15

9. Dijkstra's Algorithm cannot be applied on which of the following?
   A. Directed and weighted graphs
   B. Graphs having negative weights
   C. Unweighted graphs
   D. Undirected and unweighted graphs

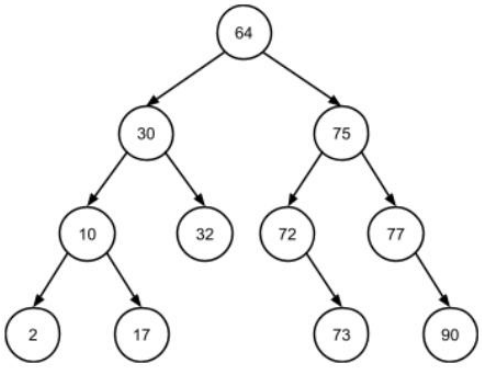
10. Given the following BST, what AVL tree rotation needs to be made to restore the balance?



   A. Right rotation around node 5, then left rotation around node 2
   B. Left rotation around node 3, then right rotation around node 5
   C. Right rotation around node 2, then left rotation around node 5
   D. Left rotation around node 5, then right rotation around node 3

11. What is the time complexity of building a heap from an arbitrary array?
   A. O (1)
   B. O (n)
   C. O (log n)
   D. O (n log n)

Question 12-14 are based on the following BST.

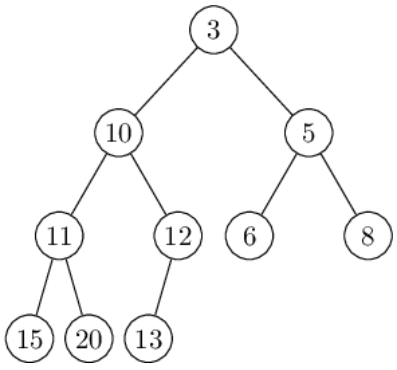


12. What is the order to search for element 72?
    A. 73 → 72
    B. 64 → 75 → 72
    C. 64 → 30 → 75 → 10 → 32 → 72
    D. 64 → 75 → 77 → 72

13. Where should we insert the element 64?
    A. The element 64 cannot be inserted since it is already in the tree
    B. The right child of element 32
    C. The left child of element 72
    D. The left child of element 73

14. If the element 30 is removed, which node should it be replaced by (no re-balancing)?
    A. element 10
    B. element 2
    C. element 32
    D. element 64

Given the following min heap, answer questions 15-18. **Each question is independent.**



15. When **`remove_first()`** is called, which node is removed?
    A. node 13
    B. node 3
    C. node 20
    D. node 8

16. When the root node 3 is removed, which node should it be replaced by before fixing the heap?
    A. node 13
    B. node 6
    C. node 5
    D. node 10

17. After the root node 3 is removed, how many percolations does the replacement node need to take in order to fix the heap?
    A. 0
    B. 1
    C. 2
    D. 3

18. When representing the above heap using an array, what's the array index of the left child of node 20?
    A. 15
    B. 16
    C. 17
    D. 18

Question 19-20 are based on the following array:

10	8	6	15	30	1	7	4

19. When converting the array above to a min heap, what is the index of the first non-leaf element?
    A. 6
    B. 5
    C. 4
    D. 3

20. What is the array-based min heap representation of the array above, using correct percolations?
    A.

1	4	6	8	30	10	7	15

    B.

1	4	6	7	8	10	15	30

    C.

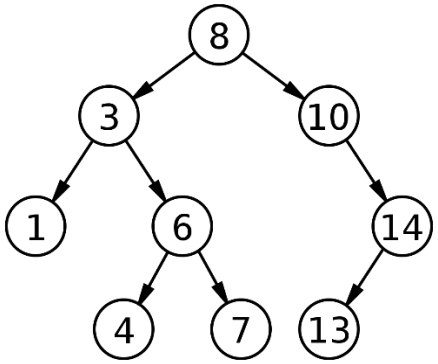
1	4	6	15	30	10	7	8

    D.

30	15	10	8	7	6	4	1

Part III: Short answers. Put your answers on the answer sheet (28 pts)

1. (8 pts, 2 pts each) Given the following binary search tree, perform the tree traversal:



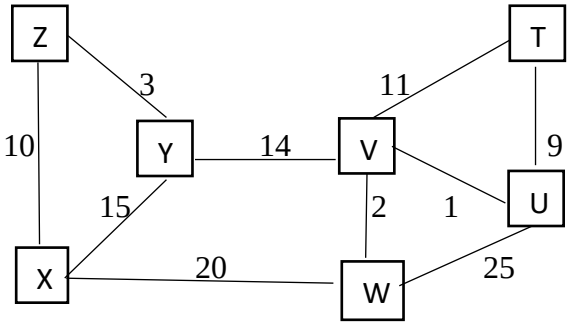
Pre-Order:

In-order:

Post-order:

Level-order:

2. (6 pts) Given the following undirected graph, using Dijkstra's algorithm and list the cost of the shortest path from X to every other vertex in the graph.



Z:

Y:

W:

V:

U

T:

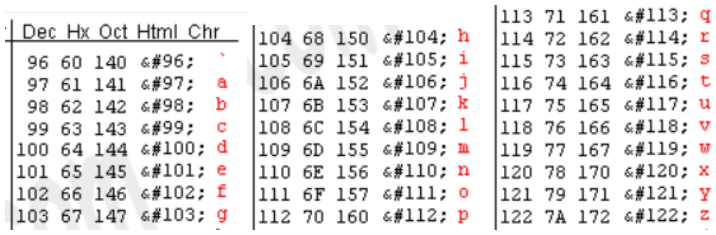
For Question 3 and 4, you are given a partly-filled hash table and a sequence of operations. Draw (in the 3rd column) the hash table that results after performing all of the operations in the sequence in the order given. In addition, indicate (in the 4th column) the load factor of the resulting table after all of the operations are performed. For each operation, only the key is provided. You may ignore the corresponding values for this problem and insert only the keys into the hash table.

For each operation in question 2 and 3, use the following hash function

```
int hash_fn(char* key) {
     int hash_val = key[0] - 'a';
     return hash_val % 10;
}
```

This function simply returns the alphabet position of the first letter of the key, starting with 0, mod 10. In other words, keys starting with ‘a’ hash to index 0, keys starting with ‘b’ hash to index 1, and so forth. You may assume all lower-case keys.

ASCII table:



Dec	Hx	Oct	Html	Chr
96	60	140	`	`
97	61	141	a	a
98	62	142	b	b
99	63	143	c	c
100	64	144	d	d
101	65	145	e	e
102	66	146	f	f
103	67	147	g	g
104	68	150	h	h
105	69	151	i	i
106	6A	152	j	j
107	6B	153	k	k
108	6C	154	l	l
109	6D	155	m	m
110	6E	156	n	n
111	6F	157	o	o
112	70	160	p	p
113	71	161	q	q
114	72	162	r	r
115	73	163	s	s
116	74	164	t	t
117	75	165	u	u
118	76	166	v	v
119	77	167	w	w
120	78	170	x	x
121	79	171	y	y
122	7A	172	z	z

3. (6 pts) For this question, use open addressing and simple linear probing to resolve collisions. Use the value __TS__ to indicate a tombstone. Tombstone values should be excluded from the load factor computation.

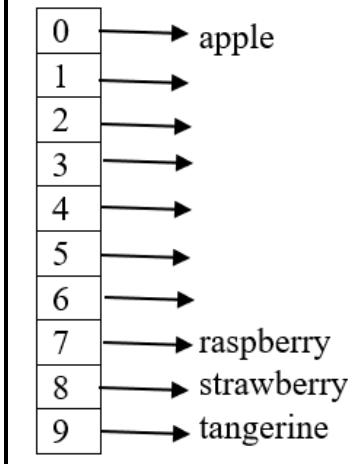
0	apple
1	
2	
3	
4	
5	
6	
7	raspberry
8	strawberry
9	tangerine

```
insert("grape");
insert("tomato");
insert("pear");
remove("apple");
```

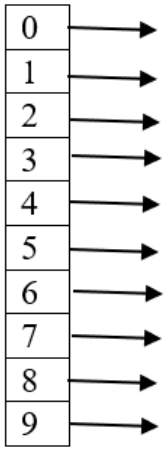
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Load factor:

4. (8 pts) For this question, use chaining to resolve collisions. **Assuming the element is inserted to the front of the list when chaining.**




```
insert("grape");
insert("tomato");
insert("pear");
remove("apple");
insert("apricots");
insert("banana");
insert("blueberry");
insert("lemon");
```



Load factor:

Extra Credit: Put your answers on the answer sheet

1. (1 pt) The teaching team of this course is never gonna _____
   A. give you up.
   B. let you down.
   C. make you cry.
   D. all of the above
   E. none of the above
   F. others: _____

2. First person to read this, stand up and yell your favorite data structure to the whole class will receive 3 extra credits on this exam.

3. (2 pts): What data structures have you learned/implemented in this course? Write them down in order: