I2206 E – INFO205
Data Structures
Dr. Zein Al Abidin IBRAHIM

Question I: Multiple Choices

1. In a heap, the index of the right child of an element located at index 23 is:

a. 11	b. 46
c. 47	d. 48

2. The most efficient implementation of a heap is using:

a. Arrays	b. Linked Lists
c. Stacks	d. Queues

3. Among the following sorting algorithms which one is the worst (time and space) in sorting an array of 100 million elements?

a. Selection Sort	b. Insertion Sort
c. Bubble Sort	d. All of them

4. The complexity of finding an element in a binary tree is ?

a. $O(n^2)$	b. $O(n)$
c. $O(\log(n))$	d. $O(1)$

5. ____is a collision-resolution strategy that searches the hash table itself, starting from the original location specified by the hash function, for an unoccupied location.

a. Linear probing	b. Open addressing
c. Quadratic probing	d. Closed addressing

6. The load factor of a table is calculated as follows:

a. table size + current number of table items	b. table size – current number of table items
c. current number of table items * table size	d. current number of table items / table size

7. Which of the following statement(s) is not TRUE?
   a- A hash function takes a message of arbitrary length and generates a fixed length code.
   b- A hash function takes a message of fixed length and generates a code of variable length.
   c- A hash function may give the same hash value for distinct messages.

a. a	b. b
c. c	d. a and c

8. What is the complexity of deleting an element in a heap tree.

a. $O(n^2)$	b. $O(n)$
c. $O(\log(n))$	d. $O(1)$

9. Which of the following represents a valid binary max heap:

a. 8 10 12 25 14 17	b. 12 10 8 25 14 17
c. 25 17 14 12 10 8	d. 14 17 25 10 12 8

10. Quick sort algorithm uses which of the following method to implement sorting?

a. Selection	b. Merging
c. Partitioning	d. Splitting

11. One similarity between a queue and a stack is:

a. Both require linked lists to be implemented.	b. Both require arrays to be implemented.
c. Both have the same complexity of operations	d. Both have the same structure.

12. In a min-heap, which of the following is correct?

a. The value of a node is less than all its children	b. The value of a node is greater than all its children

c. The Value of a node is greater than the ones in the left subtree and less than the ones in the right subtree	d. None of the choices

13. Which of the following traversals prints the data in ascending order in a binary search tree?

a. In-order traversal	b. Post-order traversal
c. Pre-order traversal	d. Level-order traversal
e. All of them	f. None of them

14. Out of the following, what is the least desirable we are waiting from a hash function?

a. it must occupy less space	b. it must cause less collisions
c. it must cause more collisions	d. it must be easy to implement

15. Which of the following is not a linear data structure?

a. Binary Tree	b. Graph
c. Heap Tree	d. All of them

16. What is the average time complexity of removing an element from a binary tree?

|	
a. O(1)	b. O(log(n))
c. O(n)	d. O(sqrt(n))

17. Which of the following represents the In-order Traversal of a Binary Tree?

a. Right -> Root -> Left	b. Left -> Root -> Right
c. Left -> Right -> Root	d. Right -> Left -> Root

18. Which of the following data structures can be used to implement queues?

a. Arrays	b. Linked lists
c. Stacks	d. All of them

19. Given the following two trees. Fill in the post-order traversal of the left tree and the in-order traversal of the right one.
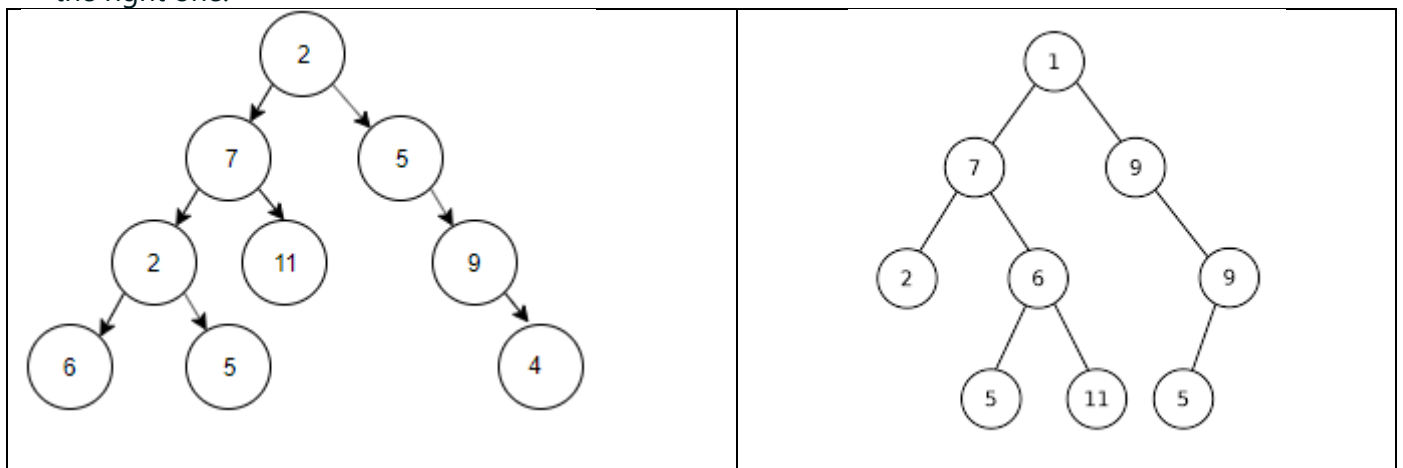
Post-order:

In-order:

20. Which data structure is used during level-order traversal of Tree?

a. Stack	b. Queue
c. Tree	d. None of them

21. Which of the following is not true about queues?

a. Head of the queue contains the last inserted element	b. Arrays can be used to implement the queue
c. Elements are stored in a sequential manner	d. Queues follows FIFO

22. What is the average case complexity of searching an element in a hash table if we use one of the closed addressing (separate chaining) method for collision resolution?

a. O(n)	b. O(log(n))
c. O(1)	d. O(sqrt(n))

23. Heap is used to implement the following:

a. Sorting	b. Priority Queues
c. a and b	d. a

24. Which of the following sort algorithms are guaranteed to be O(n log n) even in the worst case?

a. Quick sort	b. Insertion sort
c. Merge sort	d. Shell sort

25. A hash table of length 14 is shown below. The following keys were inserted in the order from left to right [41, 27, 25, 55, 24, 114, 22, 23, 36] into the table. Fill in the keys in the appropriate index if Linear Probing method is used.

0	1	2	3	4	5	6	7	8	9	10	11	12	13

26. A hash table of length 14 is shown below. The following keys were inserted in the order from left to right [41, 27, 25, 55, 24, 114, 22, 23, 9] into the table. Select the keys in the appropriate index if Quadratic Probing method is used.

0	1	2	3	4	5	6	7	8	9	10	11	12	13

27. Fill in the table below a min-based heap tree (implemented as an array) after the following data is inserted into it (from left to right): **[16, 80, 32, 97, 3, 69, 70, 25, 57, 58, 77, 78]**

0	1	2	3	4	5	6	7	8	9	10	11

28. If we use the separate chaining resolution of collision method in hashing, fill in the appropriate complexity for each of the operations in the table below:

ACTIVITY	BEST CASE	AVERAGE CASE	WORST CASE
Searching			
Insertion			
Deletion			
Space Complexity			

29. From which sequence of numbers among the below is made the following binary search tree:

a. 2, 4, 5, 6, 7, 8, 8, 10, 12
b. 7, 5, 8, 6, 2, 12, 10, 8, 4
c. 7, 8, 5, 6, 4, 2, 8, 12, 10
d. 7, 5, 8, 6, 4, 2, 12, 10, 8
e. None of them
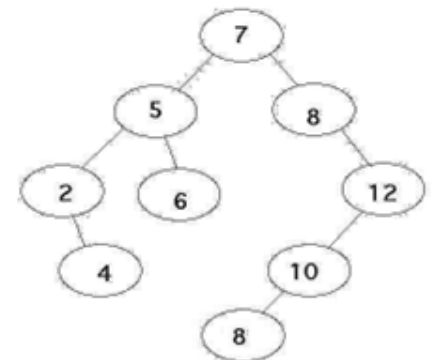
30. Given a binary search tree of n elements where you are asked to insert n more elements. The total time required for this operation is:

a. $O(n^2)$	b. $O(n\log(n))$
c. $O(n)$	d. $O(\log(n))$

31. Given a full binary tree of height M, what is the formula for determining the number of nodes in such tree? Recall that a full binary tree is a type of binary tree in which every node has either 0 or 2 child nodes.

a. 2^{M+1}	b. $2^{M-1} + 1$
c. $2^M + 1$	d. $2^{M+1} - 1$

32. Given a sorted array in descending order of n elements that we have filled from left to right into a binary search tree, what is the worst-case complexity of searching an element in the array and the tree respectively.

a. Array: O(n), Tree: O(n)	b. Array: O(n), Tree: O(log(n))
c. Array: O(log(n)), Tree: O(log(n))	d. Array: O(log(n)), Tree: O(n)

33. What is the complexity of the following piece of code? Prove that in the box below.

```
for (int x = 1; x < n*n*n; x = x * 3)
    value += (int) math.pow(x,3);
```

a. Logarithmic	b. Linear
c. Quadratic	d. Exponential

34. What is the complexity of the following piece of code? Prove that in the box below.

```
int mysterious (int n) {
    if (n == 0) return 0;
    else return mysterious (n – 1) + mysterious (n – 1);
}
```

a. Constant	b. Logarithmic
c. Linear	d. Exponential

35. What is the complexity of the following piece of code? Prove that in the box below.

```
void f(int n) {
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= i; j++)
            printf("%d ", j);
}
```

a. O(n)	b. O(log(n))
c. O(nlog(n))	d. O(n^2)

36. What is the complexity of the following piece of code? Prove that in the box below.
```
int fabilous(int n) {
    if (n <= 0)
        return 0;
    return fabilous (1) + fabilous (n-1) + n;
}
```

a. O(n)	b. O(log(n))
c. O(nlog(n))	d. O(n²)

37. What is the complexity of the following piece of code? Prove that in the box below.
```
int funct(int n){
    for(int i=1; i<=n; i++)
        for(int j=1; j<n; j+=i)
            printf("%d, %d\n",i,j);
}
```

a. O(n)	b. O(n*sqrt(n))
c. O(n²)	d. O(nlog(n))

38. Why is the efficiency of algorithms desirable?

a. Efficient algorithms always cost less money	b. Efficient algorithms always use less memory
c. An inefficient algorithm may use too much time or space	d. Inefficient algorithms always use too much memory

39. Given the content of the array [7, -12, 29, 28, 23, -8, 1, 14, 42, 15, -5, -1]. Suppose we applied two iterations of the quicksort algorithm in which the pivot (key) is always chosen as the first one, fill in the resulted array. In the second iteration, apply on the two sub-arrays.

Iteration	0	1	2	3	4	5	6	7	8	9	10	11
1												
2												

40. Given the content of the array [7, -12, 29, 28, 23, -8, 1, 14, 42, 15, -5, -1]. Suppose we applied three iterations of the bubble sort algorithm, fill in the resulted array.

Iteration	0	1	2	3	4	5	6	7	8	9	10	11
1												
2												
3												

Question II: Binary Tree

Given the following structure representing a binary tree node:

```
typedef struct BTNode{
        element data;
        struct BTNode *left, *right;
} BTNode;
```

1. Complete the following recursive function aiming to print the values in a BST that are outside the range [a, b].

```
void printOutsideRange(BTNode *root, int a, int b) {
        if (root == NULL)
            return;
        //Complete code here




}
```

2. Complete the following recursive function aiming to check if a binary tree is a sum tree. A binary sum tree is a binary tree where the value of a node is equal to the sum of the nodes present in its left subtree and right sub-tree. You should provide a solution with complexity O(n).

```
int isSumTree(BTNode *root) {
    int isSumT = 1;
    isSumTreeUtil(root, &isSumT);
    return isSumT;
}
```

```
int isSumTreeUtil(BTNode *root, int *isSumT) {
    if (root == NULL)
        return 0;
    //Complete code here




}
```

3. Complete the recursive method that takes two values v and x (x is positive) and returns v power x (vx). The complexity of the function should be no more than O(log(n)).

```
double powerPositive(double v, int x) {
    //Complete code here (x is positive only)












}
```

Pay attention, you should keep all the scratch papers stapled in the booklet exam.

Scratch