

## INTRODUCTION TO PROGRAMMING CONCEPTS, DATA STRUCTURES

Exam #2, Oct 23, 2019, Total 100 Marks

---

### Part 1: Multiple Choice (60 Marks)

---

1. What is the data structure that applies the protocol of LIFO (Last In First Out)?

- a) Hash table
- b) Stack
- c) Queue
- d) Linked list

The correct answer is [b]

2. To do breadth-first search of a tree data structure, which of the following data structure should be used?

- a) Stack
- b) Queue
- c) Either stack or queue
- d) None of the above

The correct answer is [b]

3. To do depth-first search of a tree data structure, which of the following data structure should be used?

- a) Stack
- b) Queue
- c) Either stack or queue
- d) None of the above

The correct answer is [a]

4. Which of the following algorithmic problems can be solved as a recursive function?

- a) Factorial
- b) Fibonacci numbers
- c) All the above
- d) None of the above

The correct answer is [c]

5. If you can only use stacks in your code, what is the minimum number of stacks you need to use to simulate the behavior of queue data structure?

- a) 1
- b) 2
- c) 3
- d) 4

The correct answer is [b]

6. If you can only use queues in your code, what is the minimum number of queues you need to use to simulate the behavior of stack data structure?

- a) 1
- b) 2
- c) 3
- d) 4

The correct answer is [b]

7. Which of the following statements is correct about binary trees?

- a) Each node in a binary tree must have exactly two children
- b) Each node in a binary tree must have at least two children
- c) Each node in a binary tree must have at most two children
- d) Each node in a binary tree must have more than two children

The correct answer is [c]

8. From the following running time complexity possibilities, which one is the most efficient?

- a)  $O(n)$
- b)  $O(\log n)$
- c)  $O(n^2)$  //  $n$  to the power of two
- d)  $O(2^n)$  // 2 to the power of  $n$

The correct answer is [b]

9. From the following running time complexity possibilities, which one is the least efficient?

- a)  $O(n)$
- b)  $O(\log n)$
- c)  $O(n^2)$  //  $n$  to the power of two
- d)  $O(2^n)$  // 2 to the power of  $n$

The correct answer is [d]

10) Which of the following problems can be solved in linear time (i.e.,  $O(n)$ )?

- a) Sorting an array of size  $n$
- b) Reversing an array of size  $n$
- c) All the above
- d) None of the above

The correct answer is [c]

Examples of sorting algorithms that run in linear time are counting sort, radix sort and bucket sort. Counting sort and radix sort assume that the input consists of integers in a small range. Whereas, bucket sort assumes that the input is generated by a random process that distributes elements uniformly over the interval. Sorting without assumptions can be solved in  $O(n \log n)$  time and so only b is correct. However, as the question is stated, a is also correct since technically there are many useful types of arrays that can be sorted in linear time.

11) What is the running time complexity of the following function (function\_01), assuming the size of input array is " $n$ "?

```
>>> def function_01 (array): # input array is of size n
    summation = 0
    product = 1
    for element in array:
        summation = summation + element
    for element in array:
        product = product * element
    print ("The sum of elements in the input array is: " + str(summation))
    print ("The product of elements in the input array is: " + str(product))
```

- a)  $O(\log n)$  //logarithmic runtime complexity
- b)  $O(n)$  //linear runtime complexity
- c)  $O(n^2)$  //quadratic runtime complexity
- d)  $O(2^n)$  //exponential runtime complexity

The correct answer is [b]

12) What is the running time complexity of the following function (function\_02), assuming the size of input array is "n"?

```
def function_02 (array): # input array is of size n
    for element1 in array:
        for element2 in array:
            print ("["+str(element1)+","+str(element2)+"]")
```

- a)  $O(1)$  //constant runtime complexity
- b)  $O(n)$  //linear runtime complexity
- c)  $O(n^2)$  //quadratic runtime complexity
- d)  $O(2^n)$  //exponential runtime complexity

The correct answer is [c]

13) What is the running time complexity of the following function (function\_03), assuming the size of input array is "n"?

```
>>> def function_03 (array): # input array is of size n
    i = 0
    j = len(array)-1
    while (i<j):
        temp = array[i]
        array[i] = array[j]
        array[j] = temp
        i = i+1
        j = j-1
    print (array)
```

- a)  $O(1)$  //constant runtime complexity
- b)  $O(n)$  //linear runtime complexity
- c)  $O(n^2)$  //quadratic runtime complexity
- d)  $O(2^n)$  //exponential runtime complexity

The correct answer is [b]

14) What is the running time complexity of the following function (function\_04), assuming the size of input array is "n"?

```
>>> def function_04 (array): # input array is of size n
    for element1 in array:
        for element2 in array:
            print ("["+str(element1)+","+str(element2)+"]")
    for element1 in array:
        for element2 in array:
            print(str(element1*element2))
```

- a)  $O(1)$  //constant runtime complexity
- b)  $O(n)$  //linear runtime complexity
- c)  $O(n^2)$  //quadratic runtime complexity
- d)  $O(2^n)$  //exponential runtime complexity

The correct answer is [c]

15) What is the running time complexity of the following function (function\_05), assuming the size of input array is "n"?

```
>>> def function_05 (array): # input array is of size n
    for element1 in array:
        print ("["+str(element1)+"]")
    for element1 in array:
        for element2 in array:
            print ("["+str(element1)+","+str(element2)+"]")
    for element1 in array:
        for element2 in array:
            for element3 in array:
                print ("["+str(element1)+","+str(element2)+","+str(element3)+"]")
```

- a)  $O(n)$  //linear runtime complexity
- b)  $O(n^2)$  //quadratic runtime complexity
- c)  $O(n^3)$  //cubic runtime complexity
- d)  $O(2^n)$  //exponential runtime complexity

The correct answer is [c]

-----  
Part 2: True or False (40 Marks)  
-----

1. In Python, a function must have at least one parameter.  
The above sentence is True or False? [False]
2. In Python, you can define a variable without defining its data type.  
The above sentence is True or False? [True]
3. In Python, a function must return a value at the end of the function.  
The above sentence is True or False? [False]
4. In Python, a random sequence is reproducible by default.  
The above sentence is True or False? [False]
5. In Python, a parameter of a function can be a reference to another function.  
The above sentence is True or False? [True]
6. Push and Pop operations pertain to stacks, and Enqueue and Dequeue operations pertain to hash tables.  
The above sentence is True or False? [False]
7. Hash tables must store values in either ascending or descending order.  
The above sentence is True or False? [False]
8. In a binary tree, a leaf node is a node that has exactly two children nodes.  
The above sentence is True or False? [False]
9. Finding the factorial of a given number,  $n$ , can be computed using either an iterative function or a recursive function.  
The above sentence is True or False? [True]
10. Finding the maximum number in an unsorted array of size " $n$ " can be solved in linear time (i.e.,  $O(n)$ ).  
The above sentence is True or False? [True]
11. Finding the minimum number and the maximum number in an unsorted array of size " $n$ " can be solved in linear time (i.e.,  $O(n)$ ).  
The above sentence is True or False? [True]
12. Finding the minimum number, the maximum number and the average value in an unsorted array of size " $n$ " can be solved in linear time (i.e.,  $O(n)$ ).

The above sentence is True or False? [True]

13. Using binary search to search for an element in a sorted array has linear time complexity (i.e.,  $O(n)$ ).

The above sentence is True or False? [False]

14. Searching for an element in a binary search tree has linear time complexity (i.e.,  $O(n)$ ).

The above sentence is True or False? [False] (Assuming the tree has  $n$  elements. If  $n$  is the height of the tree, this statement is true).

15. Using recursion to solve fibonacci series problem has exponential time complexity (i.e.,  $O(2^n)$ ).

The above sentence is True or False? [True]

16. When analyzing the running time complexity of an algorithm, it is correct to drop the constant factors from the analysis.

The above sentence is True or False? [True] (Unless the algorithm already runs in constant time  $O(1)$  then you can't drop the constant terms and say the algorithm runs in  $O(0)$  time).

17. When analyzing the running time complexity of an algorithm, it is correct to drop the non-dominant terms from the analysis.

The above sentence is True or False? [True]

18. If an algorithm has running time complexity of  $O(2n)$ , this means the algorithm running time complexity is actually  $O(n)$ .

The above sentence is True or False? [True]

19. If an algorithm has running time complexity of  $O(n+m)$  and  $m < n/2$ , this means the algorithm running time complexity is actually  $O(n)$ .

The above sentence is True or False? [True]

20. If an algorithm has running time complexity of  $O(n+m)$  and  $m > n$ , this means the algorithm running time complexity is actually  $O(n)$ .

The above sentence is True or False? [False]