ensia The National School of Artificial Intelligence المدرسة الوطنية العليا للذكاء الاصطناعي

Data Structure and Algorithms 2/Quiz#1

(Second Year, S3/2024-2025)

31 October 2024, 3:30 pm to 4:15 pm

Name : ..... Group : .....

I)- Multiple Choice Questions (17.5 marks)

In the following questions, each question has one correct answer. Circle your answers. (1.25) mark are awarded for each correct answer; penalty for wrong answer is (-0.5) mark.

1. What does it mean when we say that Algorithm X is asymptotically more efficient than Y ?

   a)- X will necessarily outperform Y for any small input size.

   b)- X is always slower than Y for large input sizes.

   c)- Y will always perform better than X for smaller input sizes

   d)- X is the optimal choice for every input size without exception.

   e)- X has a slower rate of time complexity growth than Y for large inputs, although it may not excel with smaller inputs.

2. What kind of list is best to answer questions such as "What is the item at position n?"

   a)- Lists implemented with an array

   b)- Doubly-linked lists.

   c)- Singly-linked lists with only head pointer

   d)- Doubly-linked or singly-linked lists are equally best.

   e)- Singly-linked lists with head and tail pointer

3. In a stack implemented using an array, what happens when you try to push an element onto a full stack?

   a)- The stack automatically resizes.

   b)- An error occurs (stack overflow).

   c)- The oldest element is removed to make space.

   d)- The new element is added at the bottom.

   e)- None of the above

4. Which of the following scenarios is best suited for using a stack?

   a)- Managing tasks in a to-do list

   b)- Maintaining order of processing requests

   c)- Undo functionality in text editors

d)- Managing print jobs in a printer.

e)- All of the above

5. if a standard queue has n elements and you want to reverse it using another queue, what will be the time complexity of this operation?

   a)- $O(1)$

   b)- $O(nlogn)$

   c)- $O(n^2)$

   d)- $O(n)$

   e)- None of the above

6. What are the time complexities of finding 8th element from beginning and 8th element from end in a singly linked list ? Let n be the number of nodes in a singly linked list, you may assume that $n > 8$.

   a)- $O(1)$ and $O(n)$

   b)- $O(1)$ and $O(1)$

   c)- $O(n)$ and $O(1)$

   d)- $O(n)$ and $O(n)$

   e)- $O(n)$ and $O(nlogn)$

7. Which of the following expressions surely supports the statement ($f(n) = \Theta(g(n))$

   a)- $f(n) \leq 4g(n)$ for all $n \geq 1$

   b)- $f(n) \geq 4g(n)$ for all $n \geq 136$

   c)- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

   d)- $3g(n) \leq f(n) \leq 4g(n)$ for all $n \geq 1$

   e)- none of the above

8. Given positive functions f(n) and g(n), if we know that $\lim_{n\to\infty}(logf(n) - logg(n)) = 1$ then we also know that

   a)- $f(n) = o(g(n))$

   b)- $f(n) = \Theta(g(n))$

   c)- $g(n) = o(f(n))$

   d)- a) and b)

   e)- more information is needed about f and g to reach a definite conclusion.

9. An algorithm takes as input an n x n Boolean matrix A. If the running time of the algorithm is $T(n) = O(nlogn)$ when n is used as the input size parameter, then which of the following expressions describes the big-Oh growth of T(m), the running time of the algorithm when $m = n^2$ is used as the size parameter?

   a)- $O(\sqrt{m}logm)$

   b)- $O(m^2logm)$

   c)- $O(mlogm)$

   d)- $O(m^2log^2m)$.

   e)- None of the above

10. What is the time complexity for this code:

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }
}
```

a)- $O(n)$

b)- $O(nlogn)$

c)- $O(n^2)$

d)- $O(n^2logn)$

e)- $O(log^2n)$

11. What will be the recurrence relation of the following code?

```
Int sum(int n)
{
   If(n==1)
       return 1;
   else
       return n+sum(n-1);
}
```

a)- $T(n) = T(n/2) + n$

b)- $T(n) = T(n-1) + n$

c)- $T(n) = T(n-1) + O(1)$

d)- $T(n) = T(n/2) + O(1)$

e)- $T(n) = 2T(n-1) + O(1)$

12. What is the running time complexity for this function:

```
int bar(n)
{ if ( n<=0)
         return 1
else
         return bar(n-1) +  bar(n-1)
}
```

a)- $O(2^n)$

b)- $O(n^2)$

c)- $O(2n)$

d)- $O(n^3)$

e)- $O(n)$

13. If T(n) satisfies $T(n) = 2T(\frac{n}{3}) + \sqrt{n}$, then

a)- $T(n) = \Theta(\sqrt{n})$

b)- $T(n) = \Theta(n^2)$

c)- $T(n) = \Theta(n^{log_3 2})$

d)- $T(n) = \Theta(n^{log 3})$

e)- $T(n) = \Theta(\sqrt{n}logn)$

14. Which of the following recurrences cannot be solved directly by the Master Theorem?

a)- $T(n) = 16T(\frac{n}{4}) + n$

b)- $T(n) = T(\frac{n}{5}) + 20$

c)- $T(n) = 3T(\frac{n}{4}) + nlogn$

d)- $T(n) = 2T(\frac{n}{2}) + nlogn$

e)- $T(n) = 16T(\frac{n}{4}) + n!$

II)- Short answer Questions (2.5 marks)

Describe the worst case running time of the following code functions in Big-Oh notation in terms of the variable n. (Showing your work is not required)

```
void fct1(int n) {
    for (int i = n * n; i > 0; i--) {
        for (int k = 0; k < n; ++k) {
            std::cout << "k = " << k << std::endl;
        }
        for (int j = 0; j < i; ++j) {
            std::cout << "j = " << j << std::endl;
        }
        for (int m = 0; m < 5000; ++m) {
            std::cout << "m = " << m << std::endl;
        }
    }
}
}
```

```
void fct2(int n) {
    for (int i = 0; i < 2 * n; ++i) {
        int j = 0;
        while (j < n) {
            std::cout << "j = " << j << std::endl;
            j = j + 5;
        }
    }
}
```

Quiz#1 Answers

I)-Multiple Choice Questions

Question	Answer
1	e
2	a
3	b
4	c
5	c
6	a
7	d
8	b
9	a
10	b
11	c
12	a
13	c
14	d

II)- Short answer Questions

fct1 is $O(n^4)$

fct2 is $O(n^2)$