

GATE 2022

Q1. An algorithm is made up of 2 modules M_1 and M_2 . If order of M_1 is $f(n)$ and M_2 is $g(n)$ then the order of the algorithm is

- (a) $\max(f(n), g(n))$
- (b) $\min(f(n), g(n))$
- (c) $f(n) + g(n)$
- (d) $f(n) * g(n)$

Q2. The concept of order (Big O) is important because

- (a) it can be used to decide the best algorithm that solves a given problem
- (b) it determines the maximum size of a problem that can be solved in a given system, in a given amount of time
- (c) it is the lower bound of the growth rate of the algorithm
- (d) Both (a) and (b)

Q3. There are 4 different algorithms. A1, A2, A3, A4 to solve a given problem with the order $\log(n)$, $\log(\log(n))$, $n\log(n)$, $n/\log(n)$ respectively. Which is the best algorithm?

- (a) A1
- (b) A2
- (c) A3
- (d) A4

Q4. Consider the following 2 functions:

$$f(n) = n^3, \text{ if } 0 \leq n < 10,000$$

$$= n^2, \text{ otherwise}$$

$$g(n) = n, \text{ if } 0 \leq n < 100$$

$$= n^2 + 5n, \text{ otherwise}$$

Which of the following option is correct?

(a) $f(n)$ is $O(n^3)$

(b) $g(n)$ is $O(n^3)$

(c) $O(f(n))$ is same as $O(g(n))$

(d) $g(n)$ is $O(1)$

Q5. What is time complexity of fun()?

```
int fun(int n)
{
    int count=0;
    for (int i= n; i> 0; i/=2)
        for(int j=0; j<i; j++)
            count+=1;
    return count;
}
```

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

Q6. What is the time complexity of fun()?

```
int fun(int n)
{
    int count=0;
    for (int i=0; i<n; i++)
        for (int j=i; j>0; j--)
            count= count+1;
    return count;
}
```

- (a) $\Theta(n)$
- (b) $\Theta(n^2)$
- (c) $\Theta(n \log n)$
- (d) $\Theta(n \log n \log n)$

Q7. Let $w(n)$ and $A(n)$ denote respectively, the worst case and average case running time of an algorithm executed on an input of size n . which of the following is ALWAYS TRUE?

- (a) $A(n) = \Omega(n)$
- (b) $A(n) = \theta(n)$
- (c) $A(n) = O(n)$
- (d) $A(n) = o(n)$

Q8. Which of the following is not $O(n^2)$?

(a) $(15^{10}) * n + 12099$

(b) $n^{1.98}$

(c) $n^3 / (\text{sqrt}(n))$

(d) $(2^{20}) * n$

Q9. Which of the given options provides the increasing order of asymptotic complexity of functions f_1 , f_2 , f_3 and f_4 ?

$$f_1(n) = 2^n$$

$$f_2(n) = n^{(3/2)}$$

$$f_3(n) = n \log n$$

$$f_4(n) = n^{(\log n)}$$

(a) f_3, f_2, f_4, f_1

(b) f_3, f_2, f_1, f_4

(c) f_2, f_3, f_1, f_4

(d) f_2, f_3, f_4, f_1

Q10. In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, same set of statements are executed. Consider the following for loops:

(1) `for(i = 0; i < n; i++)`

(2) `for(i = 0; i < n; i += 2)`

(3) `for(i = 1; i < n; i *= 2)`

(4) `for(i = n; i > -1; i /= 2)`

If n is the size of input(positive), which function is most efficient(if the task to be performed is not an issue)?

(a) 1

(b) 2

(c) 3

(d) 4

Q11. Consider the following functions:

$$f(n) = 3n^{\sqrt{x}}$$

$$g(n) = 2^{\sqrt{x \log_2 n}}$$

$$h(n) = n!$$

Which of the following is true?

- (a) $h(n)$ is $O(f(n))$
- (b) $h(n)$ is $O(g(n))$
- (c) $g(n)$ is not $O(f(n))$
- (d) $f(n)$ is $O(g(n))$

Solutions:

- 1) A
- 2) D
- 3) B
- 4) C
- 5) C
- 6) B
- 7) C
- 8) C
- 9) A
- 10) C
- 11) D