Password: wia2005test

1	True or false: Let f be a function, if f = O(g) and g = O(h), then f = O(h). Select one: ◉ True ○ False
2	True or false Let f be a function, if f = Ω (g) and h = Ω (g), then f = Ω (h). Select one: ○ True ◉ False
3	True or false A function with a faster growth rate is better than a function with a slower growth rate. Select one: ○ True ◉ False

4

Describe the functions in Figure 1:

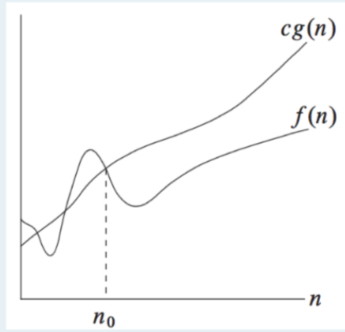

Figure 1

Select one:

- a. Function g(n) is a subset of Big O of f(n), if there is a positive constant n_0 and c such that at and to the right of n_0 , the values of f(n) lies on or below cg(n).
- b. Function f(n) is a subset of Big Omega of g(n), if there is a positive constant n_0 and c such that at and to the right of n_0 , the values of f(n) above cg(n).
- c. Function f(n) is a subset of Big Omega of g(n), if there is a positive constant n_0 and c such that at and to the right of n_0 , the values of g(n) lies on or below cg(n).
- d. Function f(n) is a subset of Big O of g(n), if there is a positive constant n_0 and c such that at and to the right of n_0 , the values of f(n) lies on or below cg(n).

Clear my choice

5

Compare the following functions and select the case based on the order of growth rate:

f(n) = $\sqrt{2}$n

g(n) = n^2 log n

- a. f(n) = Ω(g(n))
- b. g(n) = O(f(n))
- c. f(n) = O(g(n))
- d. f(n) = θ(g(n))

Clear my choice

6	Compare the following functions and select the case based on the order of growth rate: $f(n) = 2^{\sqrt{\log n}}$ $g(n) = n^{4/3}$ (•) a. $f(n) = O(g(n))$ ( ) b. $g(n) = O(f(n))$ ( ) c. $f(n) = \Omega(g(n))$ ( ) d. $f(n) = \theta(g(n))$ Clear my choice
7	Given $f(n) = n^2 – 5$, $g(n) = n^3 + 7$ and $n_0 = -2$, define the case based on the order of growth rate and based on the definition, state the possible value of n and c that proves that it is the case. (•) a. $f(n) = O(g(n))$, $n = 1$, $c = 1$ ( ) b. $f(n) = O(g(n))$, $n = 1$, $c = -2$ ( ) c. $f(n) = O(g(n))$, $n = -2$, $c =1$ ( ) d. $f(n) = \Omega(g(n))$, $n = 1$, $c = 1$ Clear my choice

8	Given $f(n) = n^2$, $g(n) = \log n + 5$ and $n_0 = 2.3$, define the case based on the order of growth rate and based on the definition, state the possible value of n and c that proves that it is the case. ○ a. $f(n) = \Omega(g(n))$, n = 1, c = -1 ○ b. $f(n) = O(g(n))$, n = 3, c = 1 ◉ c. $g(n) = O(f(n))$, n = 3, c = 1 ○ d. $g(n) = O(f(n))$, n = 1, c =1 Clear my choice

9	Calculate the running time complexity of Function A.
	
	`Function_A()`
	`max = a_i`
	`For i = 2 to n`
	`      If a_i > max then`
	`            Set max = a_i`
	`      Endif`
	`Endfor`
	
	○ a. O(n log n)
	○ b. O(n-2)
	○ c. $O(n^2)$
	◉ d. O(n)
	
	Clear my choice

10	Calculate the running time complexity of Function B.
11	Write the recurrence relation for Algorithm A.

10

Calculate the running time complexity of Function B.

```
Function_B(arr)
n = len(arr)
For i in range(n):
     For j in range(0, n-i-1):
          If arr[j] > arr[j+1]:
               arr[j], arr[j+1] = arr[j+1], arr[j]
          Endif
     Endfor
Endfor
```

- ○ a. O(2n)
- ○ b. O(n)
- ◉ c. $O(n^2)$
- ○ d. O(n-1)

Clear my choice

11

Write the recurrence relation for Algorithm A.

```
Algorithm_A(n):
  if n == 0:
    return 0
  else:
    return 2 * Algorithm_A(n/2) + n
```

- ○ a. T(n/2) = 2T(n/2) + n
- ○ b. T(n) = T(n/2) – n
- ○ c. T(n) = 2T(n) + (n/2)
- ◉ d. T(n) = 2T(n/2) + n

Clear my choice

12	Using the back substitution method, guess the term when T(n/2) for Algorithm A in Question 11. a. $T(n/2) = T(n/4) + n/2$ b. $T(n/2) = 2T(n) + (n/2)$ c. $T(n/2) = 4T(n/4) + 2n$ d. $T(n/2) = 2T(n/2) + n$ Clear my choice Not sure answer , feel weird Answer got bug $T(n/2) = 2T(n/4) + n/2$
13	Using the back substitution method, guess the last term for Algorithm A in Question 11. a. $T(n) = 2T(n/2^k) + kn$ b. $T(n) = 2^kT(n/2^k) + n$ c. $T(n) = 2^kT(n/2^k) + kn$ d. $T(n) = 2^kT(n/2) + n$ Clear my choice
14	$T(n)=O(n \log n)$ Yess!!

<table>
<tr>
<td></td>
<td>
Using the back substitution method, given the base case T(1) = 1, calculate the running time complexity of Algorithm A in Question 11.

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

Clear my choice
</td>
</tr>
<tr>
<td>15</td>
<td>
$(3/4^2)cn^2$?

$(3/4^2)cn^2$ ~~

Given a recurrence relation $T(n) = 3T(n/4) + cn^2$, using the recursion tree method, calculate the level sum when $T(n/4)$.

a. $(3/4)cn$
b. $(3/4)cn^2$
c. $(3/4^2)cn^2$
d. $(3/4^2)cn$

Clear my choice
</td>
</tr>
<tr>
<td>16</td>
<td>
$k = \log_4 n$

$k = \log_4 n$ ~~
</td>
</tr>
</table>

Given a recurrence relation in Question 15, calculate the height of the recursion tree, k.

a. $k = \log_{3/4} n$

b. $k = \log_3 n$

c. $k = \log_2 n$

d. $k = \log_4 n$

Clear my choice

17

Case 3: If $\log_b a < k$ then

a) If $p \geq 0$, then, $T(n) = \Theta(n^k \log^p n)$

b) If $p < 0$, then, $T(n) = O(n^k)$

Using the master method, solve T(n) = 3T(n/4) + n.

a. $\Theta(n^2)$

b. $\Theta(n)$

c. $\Theta(n^{\log_2 3})$

d. $\Theta(n \log n)$

Clear my choice

18	Using the master method, solve $T(n) = 7T(n/49) + n^2 \log n$. a. $\Theta(n)$ b. $\Theta(n^2)$ c. $\Theta(n \log n)$ d. $\Theta(n^2 \log n)$ Clear my choice
19	Jsing the master method, solve $T(n) = 4T(n/2) + n^2$. a. $\Theta(n)$ b. $\Theta(n^2)$ c. $\Theta(n \log n)$ d. $\Theta(n^2 \log n)$ Clear my choice
20	Using the master method, solve $T(n) = 16T(n/4) + n$. a. $\Theta(n \log n)$ b. $\Theta(n)$ c. $\Theta(n^2 \log n)$ d. $\Theta(n^2)$ Clear my choice