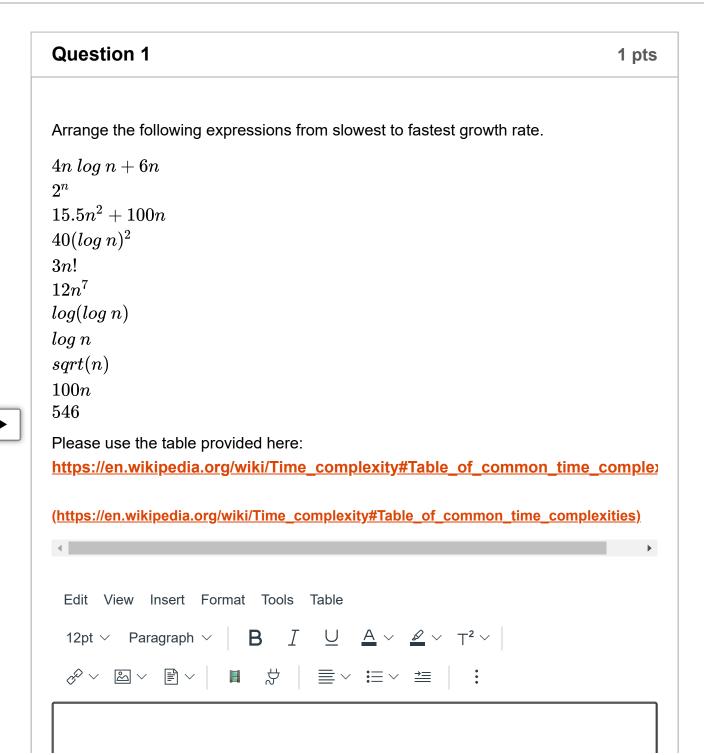
Practice Quiz - Big O Complexity

Started: Apr 8 at 2:13pm

Quiz Instructions

This quiz is not an actual quiz and will not be counted toward the final grade. It is provided to test whether you have grasped the topic.

Be sure to perform the required readings assigned on this topic.



p (i) 0 words | </> [1] | | |

Question 2 1 pts

Determine the big O complexity for the following function, which represents the number of steps required for some algorithm.

$$F(n)=2\sqrt{n}+5n*log\,n+100n^2$$

- $\bigcirc O(100n^2)$
- $\bigcirc O(n)$
- $\bigcirc O(n \log n)$
- $\bigcirc O(n^2)$

Question 3 1 pts

Determine the big O complexity for the following function, which represents the number of steps required for some algorithm.

$$T(n) = 3(2^n) + n^8 + 1024n$$

$\bigcirc~O(3(2^n))$			
$\bigcirc~O(2^n)$			
$\bigcirc \ O(n^8)$			
$\bigcirc \ O(n)$			

Question 4 1 pts

Determine the big O complexity for the following function, which represents the number of steps required for some algorithm.

$$G(n)=3log\ n!+54log(log\ n)+20(log\ n)^2$$

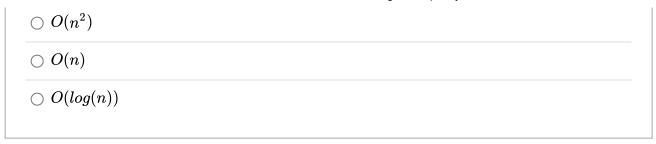
- $\bigcirc \ O(log(log\ n))$
- $\bigcirc O(\log n)$
- $\bigcirc\ O(n\ log\ n)$
- $\bigcirc \ O((\log n)^2)$

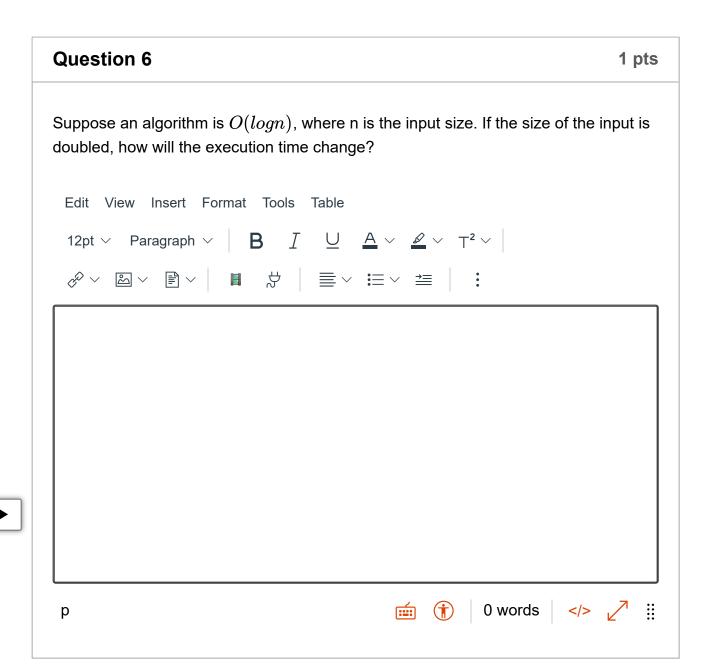
Question 5 1 pts

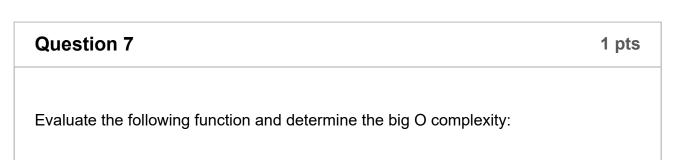
Evaluate the following code segment and determine the big O complexity:

```
def sum(n):
    sum = 0
    i = n
    while i > 0:
        sum += i
        i = i // 2
    return sum
```

 $\bigcirc O(1)$







Question 8 1 pts

Evaluate the following code segment and determine the big O complexity:

```
test = 0
for i in range(n):
    test = test + 1

for j in range(n):
    test = test - 1
```

- $\bigcirc O(n^2)$
- $\bigcirc O(1)$
- $\bigcirc \ O(log(n))$
- $\bigcirc O(n)$

Question 9 1 pts

Evaluate the following code segment and determine the big O complexity:

```
for i in range( n ) :
    if i % 3 == 0 :
        sum = 0
        j = n
        while j > 0 :
            sum += j
            j = j // 2
    elif i % 2 == 0 :
        for j in range( 5 ) :
            sum += j
    else :
        for j in range( n ) :
            sum += j
```

- $\bigcirc O(n^2)$
- $\bigcirc O(n * log(n))$
- $\bigcirc O(n^3)$
- $\bigcirc O(n)$

Question 10 1 pts



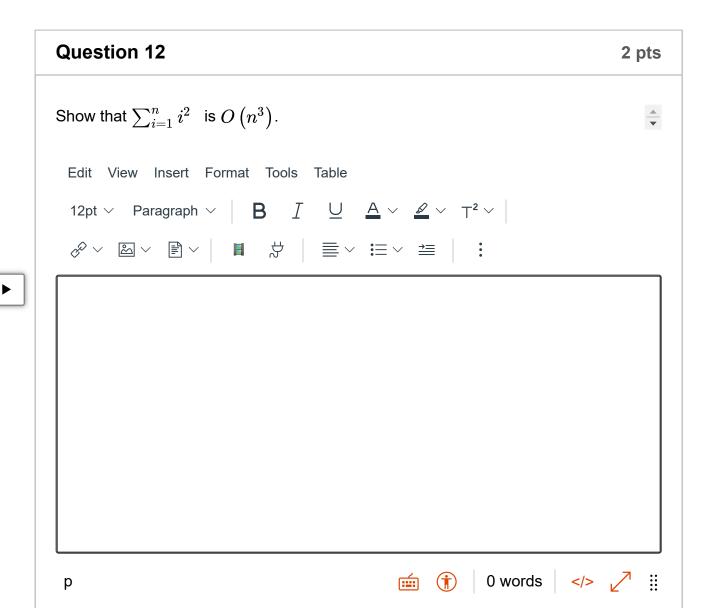
Determine the big O complexity of the following function:

```
def example( n ):
    count = 0
    for i in range( n ) :
        for j in range( 125 ) :
            count += 1
    return count
```

- $\bigcirc \ O(n^2)$
- $\bigcirc O(n)$
- $\bigcirc O(1)$

 $\bigcirc O(125*n)$

Question 11	1 pts		
A program that uses an $O(n^2)$ algorithm will always take longer to run than a program that uses an $O(n \ log \ n)$ algorithm.			
○ True			
○ False			



Not saved Submit Quiz

•