

**Started on** Thursday, 20 March 2025, 10:31 AM**State** Finished**Completed on** Thursday, 20 March 2025, 10:45 AM**Time taken** 13 mins 53 secs**Marks** 5.00/15.00**Grade** 33.33 out of 100.00**Question 1**

Complete

Mark 0.00 out of 1.00

```
for (int i = 1; i <= n; i *= 2) {  
    for (int j = 1; j <= i; j++) {  
        // Constant time operation  
    }  
}
```

What is the time complexity of the above code?

- ☐ a.  $O(n \log n)$
- ☒ b.  $O(n^2)$
- ☐ c.  $O(\log^2 n)$
- ☐ d.  $O(\text{👎})$

**Question 2**

Complete

Mark 1.00 out of 1.00

```
for (int i = 1; i <= n; i++) {  
    for (int j = 1; j <= sqrt(👎) ; j++) {  
        // Constant time operation  
    }  
}
```

What is the time complexity?

- ☐ a.  $O(\text{👎})$
- ☒ b.  $O(n \sqrt{\text{👎}})$
- ☐ c.  $O(n \log n)$
- ☐ d.  $O(n^2)$

**Question 3**

Complete

Mark 0.00 out of 1.00

```
void recursive(int n) {  
    if (n <= 1) return;  
    recursive(n/2);  
    recursive(n/2);  
}
```

What is the time complexity?

- ☐ a.  $O(n \log n)$
- ☒ b.  $O(\text{👎})$
- ☐ c.  $O(2^{\log n})$
- ☐ d.  $O(\log n)$

**Question 4**

Complete

Mark 0.00 out of 1.00

```
for (int i = 1; i < n; i *= 3) {  
    // Constant time operation  
}
```

What is the time complexity?

- ☐ a.  $O(n^2)$
- ☐ b.  $O(n \log n)$
- ☒ c.  $O(\text{👎})$
- ☐ d.  $O(\log n)$

**Question 5**

Complete

Mark 1.00 out of 1.00

```
void recur(int n) {  
    if (n <= 1) return;  
    recur(n/3);  
    recur(n/3);  
    recur(n/3);  
}
```

What is the time complexity?

- ☐ a.  $O(\log n)$
- ☐ b.  $O(n \log n)$
- ☒ c.  $O(3^{\log n})$
- ☐ d.  $O$  🙅

**Question 6**

Complete

Mark 0.00 out of 1.00

```
for (int i = n; i > 1; i /= 2) {  
    for (int j = 1; j <= i; j++) {  
        // Constant time operation  
    }  
}
```

What is the time complexity?

- ☐ a.  $O(\log n)$
- ☐ b.  $O(n^2)$
- ☐ c.  $O(n \log n)$
- ☒ d.  $O$  🙅

**Question 7**

Complete

Mark 0.00 out of 1.00

```
int fib(int n) {  
    if (n <= 1) return 1;  
    return fib(n-1) + fib(n-2);  
}
```

What is the time complexity?

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

**Question 8**

Complete

Mark 0.00 out of 1.00

```
for (int i = 1; i <= n; i *= 2) {  
    for (int j = i; j <= n; j++) {  
        // Constant time operation  
    }  
}
```

What is the time complexity?

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

**Question 9**

Complete

Mark 0.00 out of 1.00

```
void recurse(int n) {  
    if (n <= 1) return;  
    recurse(n/2);  
    recurse(n/3);  
}
```

What is the time complexity?

- ☒ a.  $O(2^n)$
- ☐ b.  $O(\log n)$
- ☐ c.  $O(n \log n)$
- ☐ d.  $O$  🗑️

**Question 10**

Complete

Mark 1.00 out of 1.00

```
for (int i = 1; i <= n; i++) {  
    for (int j = i; j <= n; j++) {  
        // Constant time operation  
    }  
}
```

What is the time complexity?

- ☐ a.  $O(n \log n)$
- ☐ b.  $O(n^3)$
- ☐ c.  $O$  🗑️
- ☒ d.  $O(n^2)$

**Question 11**

Complete

Mark 0.00 out of 1.00

```
for (int i = 1; i <= n; i++) {  
    for (int j = 1; j <= i; j++) {  
        // Constant time operation  
    }  
}
```

What is the time complexity?

- ☐ a.  $O(n^2)$
- ☒ b.  $O(n \log n)$
- ☐ c.  $O(n^3)$
- ☐ d.  $O$  🙄

**Question 12**

Complete

Mark 0.00 out of 1.00

```
void divRecur(int n) {  
    if (n <= 1) return;  
    divRecur(n / 4);  
}
```

What is the time complexity?

- ☒ a.  $O(2^n)$
- ☐ b.  $O(\log n)$
- ☐ c.  $O$  🙄
- ☐ d.  $O(n \log n)$

**Question 13**

Complete

Mark 0.00 out of 1.00

```
for (int i = 1; i < n; i *= 2) {  
    for (int j = i; j < n; j += i) {  
        // Constant time operation  
    }  
}
```

What is the time complexity?

- ☐ a.  $O(n \log n)$
- ☒ b.  $O(n^2)$
- ☐ c.  $O$  🗨
- ☐ d.  $O(\log n)$

**Question 14**

Complete

Mark 1.00 out of 1.00

```
void expRecur(int n) {  
    if (n <= 1) return;  
    expRecur(n-1);  
    expRecur(n-1);  
}
```

What is the time complexity?

- ☐ a.  $O(\log n)$
- ☐ b.  $O$  🗨
- ☐ c.  $O(n \log n)$
- ☒ d.  $O(2^n)$

**Question 15**

Complete

Mark 1.00 out of 1.00

```
for (int i = 1; i < n; i *= 2) {  
    for (int j = i; j < n; j *= 2) {  
        // Constant time operation  
    }  
}
```

- ☐ a.  $O(n^2)$
- ☒ b.  $O(\log^2 n)$
- ☐ c.  $O$  🗨
- ☐ d.  $O(n \log n)$

