Question 1

Not yet answered

Marked 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👎
- b. O(n^2)
- c. O(n log n)
- d. O(log^2 n)

Question 2

Not yet answered

Marked 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(n sqrt👎)
- b. O(n log n)
- c. O(n^2)
- d. O👎

Question 3

Not yet answered

Marked 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👎
- b. O(n log n)
- c. O(log n)
- d. O(2^log n)

Question 4

Not yet answered

Marked out of 1.00

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

What is the time complexity?

- a. O👎
- b. O(n log n)
- c. O(n^2)
- d. O(log n)

Question 5

Not yet answered

Marked 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(n log n)
- b. O👎
- c. O(3^log n)
- d. O(log n)

Question 6

Not yet answered

Marked 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(n log n)
- b. O👎
- c. O(n^2)
- d. O(log n)

Question 7

Not yet answered

Marked 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👎
- b. O(log n)
- c. O(n^2)
- d. O(2^n)

Question 8

Not yet answered

Marked 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^2)
- b. O(n log n)
- c. O👎
- d. O(log n)

Question 9

Not yet answered

Marked 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(log n)
- b. O(n log n)
- c. O👎
- d. O(2^n)

Question 10

Not yet answered

Marked 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^2)
- c. O👎
- d. O(n^3)

Question 11

Not yet answered

Marked 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^3)
- b. O(n^2)
- c. O👎
- d. O(n log n)

Question 12

Not yet answered

Marked out of 1.00

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

What is the time complexity?

- a. O👎
- b. O(log n)
- c. O(n log n)
- d. O(2^n)

Question 13

Not yet answered

Marked 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(log n)
- b. O(n log n)
- c. O(n^2)
- d. O👎

Question 14

Not yet answered

Marked 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(n log n)
- b. O(log n)
- c. O👎
- d. O(2^n)

Question 15

Not yet answered

Marked 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(n log n)
- d. O🤜