

Complexity Questions

ComplexCode1

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {  
    for (int i = 0; i < N; i++) {  
        System.out.println("Hey");  
  
        if (i == 5) {  
            i = N;  
        }  
    }  
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(1)$

ComplexCode2

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {  
    for (int i = 1; i < N; i *= 2) {  
        System.out.println("Hey");  
    }  
  
    for (int i = 0; i < N; i += 2) {  
        System.out.println("You");  
    }  
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N)$

ComplexCode3

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {  
    for (int i = N; i > N - 5; i--) {  
        for (int j = 0; j < N / 2; j++) {  
            for (int k = 0; k < N * 3; k++) {  
                System.out.println("Hey");  
            }  
        }  
    }  
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N^2)$

ComplexCode4

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {  
    for (int i = 1; i < N; i *= 2) {  
        System.out.println("Hey");  
  
        for (int j = 0; j < N; j += 2) {  
            System.out.println("You");  
        }  
    }  
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$

- $O(N!)$

Correct answer:

$O(N \log N)$

ComplexCode5

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {
    for (int i = N; i > 0; i--) {
        for (int j = 100; j < N; j++) {
            for (int k = 15; k < N * 2; k += 5) {
                System.out.println("Hey");
            }
        }
    }
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N^3)$

ComplexCode6

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {
    int k = 0;
    for (int i = 1; i <= N; i++) {
        for (int j = 1; j <= i; j++) {
            k++;
        }
    }
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$

- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N^2)$

ComplexCode7

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {
    for (int i = N / 2; i > 0; i--) {
        if (i % 2 == 0) {
            for (int j = 0; j < N; j += 2) {
                System.out.println("Hey");
            }
        } else {
            for (int j = 1; j < N; j *= 2) {
                System.out.println("You");
            }
        }
    }
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N^2)$

ComplexCode8

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N) {
    for (int i = N; i > 0; i /= 2) {
        System.out.println("Hey");
    }

    for (int i = 1; i < N; i *= 2) {
        System.out.println("You");
    }
}
```

- $O(1)$

- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(\log N)$

ComplexCode9

What is the asymptotic complexity (big-O) of the following code?

```
public int f(int[] a, int N) {
    if (N <= 0) {
        return a[0];
    } else {
        return a[N-1] + f(a, N-1) + f(a, N-1);
    }
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(2^N)$

ComplexCode10

What is the asymptotic complexity (big-O) of the following code?

```
public void f(int N, String word) {
    if (N <= 0) {
        return;
    } else {
        System.out.println(word);
        f(N - 1, word.toLowerCase());
        f(N - 1, word.toUpperCase());
    }
}
```

- $O(1)$

- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(2^N)$

ComplexCode11

What is the asymptotic complexity (big-O) of the following code?

```
// x is the head of a linked list with N nodes
public void f(Node x) {
    Node t1 = x;
    while (t1 != null) {
        System.out.println(t1.data);
        Node t2 = t1;
        while (t2 != null) {
            System.out.println(t2.data);
            t2 = t2.next;
        }
        t1 = t1.next;
    }
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N^2)$

ComplexCode12

What is the asymptotic complexity (big-O) of the following code?

```
public int f(int N) {
    if (N <= 0) {
        return 0;
    } else {
```

```

        for (int i = 0; i < N; i++) {
            System.out.println("hey");
        }
        return N + f(N-1);
    }
}

```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N^2)$

ComplexCode13

What is the asymptotic complexity (big-O) of the following code?

```

public void f(int[] a, int N) {
    TreeMap<Integer,Integer> x = new TreeMap<Integer,Integer>();

    for (int i = 0; i < N; i++) {
        x.put(a[i], 2 * a[i]);
    }
}

```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N \log N)$

ComplexCode14

What is the asymptotic complexity (big-O) of the following code?

```

public void f(int[] a, int N) {
    HashMap<Integer,Integer> x = new HashMap<Integer,Integer>();
}

```

- ```

 for (int i = 0; i < N; i++) {
 x.put(a[i], 2 * a[i]);
 x.put(2 * a[i], x.get(a[i]));
 System.out.println(x.get(2 * a[i]));
 }
 }

```
- $O(1)$
  - $O(\log N)$
  - $O(N)$
  - $O(N \log N)$
  - $O(N^2)$
  - $O(N^3)$
  - $O(2^N)$
  - $O(N!)$

**Correct answer:**

$O(N)$

## ComplexCode15

What is the asymptotic complexity (big-O) of the following code?

- ```

public void f(int[] a, int N) {
    TreeMap<Integer,Integer> x = new TreeMap<Integer,Integer>();
    HashMap<Integer,Integer> y = new HashMap<Integer,Integer>();

    for (int i = 0; i < N; i++) {
        x.put(a[i], 2 * a[i]);
        y.put(a[i], 3 * x.get(a[i]));
    }
}

```
- $O(1)$
 - $O(\log N)$
 - $O(N)$
 - $O(N \log N)$
 - $O(N^2)$
 - $O(N^3)$
 - $O(2^N)$
 - $O(N!)$

Correct answer:

$O(N \log N)$

ComplexCode16

What is the asymptotic complexity (big-O) of the following code?

```

public void f(int[] a, int N) {

```



```

        ArrayList<Integer> x = new ArrayList<Integer>();

        x.add(a[N / 2]);
    }

```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(1)$

ComplexCode17

What is the asymptotic complexity (big-O) of the following code?

```

public void f(int[] a, int N) {
    ArrayList<Integer> x = new ArrayList<Integer>();

    for (int i = 0; i < N; i++) {
        if (!x.contains(a[i])) {
            x.add(a[i]);
        }
    }
}

```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N^2)$

ComplexCode18

What is the asymptotic complexity (big-O) of the following code?

```

public void f(int[] a, int N) {
    ArrayList<Integer> x = new ArrayList<Integer>();
    TreeMap<Integer, Integer> y = new TreeMap<Integer, Integer>();
}

```

- ```

 for (int i = 0; i < N; i++) {
 if (!y.containsKey(a[i])) {
 x.add(a[i]);
 y.put(a[i], 1);
 }
 }
 }

```
- $O(1)$
  - $O(\log N)$
  - $O(N)$
  - $O(N \log N)$
  - $O(N^2)$
  - $O(N^3)$
  - $O(2^N)$
  - $O(N!)$

**Correct answer:**

$O(N \log N)$

## ComplexCode19

What is the asymptotic complexity (big-O) of the following code?

- ```

public int f(int N) {
    if (N <= 0) {
        return 1;
    } else {
        return f(N / 2) + f(N / 2);
    }
}

```
- $O(1)$
 - $O(\log N)$
 - $O(N)$
 - $O(N \log N)$
 - $O(N^2)$
 - $O(N^3)$
 - $O(2^N)$
 - $O(N!)$

Correct answer:

$O(N)$

ComplexCode20

What is the asymptotic complexity (big-O) of the following code?

```

public int f(int N) {

```

- ```

 if (N <= 0) {
 return 1;
 } else {
 return 2 * f(N / 2);
 }
 }

```
- $O(1)$
  - $O(\log N)$
  - $O(N)$
  - $O(N \log N)$
  - $O(N^2)$
  - $O(N^3)$
  - $O(2^N)$
  - $O(N!)$

**Correct answer:**

$O(\log N)$

## ComplexCode21

What is the asymptotic complexity (big-O) of the following code?

- ```

// a[] contains N elements
public void f(int[] a, int N) {
    int count = N;

    while (count > 0) {
        a[0] = 5;
        // sort N elements in a[] using selection sort
        selectionSort(a, N);
        count--;
    }
}

```
- $O(1)$
 - $O(\log N)$
 - $O(N)$
 - $O(N \log N)$
 - $O(N^2)$
 - $O(N^3)$
 - $O(2^N)$
 - $O(N!)$

Correct answer:

$O(N^3)$

ComplexCode22

What is the asymptotic complexity (big-O) of the following code?

- ```
public void f(int[] a, int N) {
 // sort N elements in a[] using quick sort.
 // This implementation of quick sort chooses
 // always the first element as pivot
 quickSort(a, N);

 // do it again
 quickSort(a, N);
}
```
- $O(1)$
  - $O(\log N)$
  - $O(N)$
  - $O(N \log N)$
  - $O(N^2)$
  - $O(N^3)$
  - $O(2^N)$
  - $O(N!)$

**Correct answer:**

$O(N^2)$

## ComplexCode23

What is the asymptotic complexity (big-O) of the following code?

- ```
public void f(int[] a, int N) {
    if (N <= 0) {
        return;
    } else {
        a[0] = 5;
        // sort N elements in a[] using selection sort
        selectionSort(a, N);
        f(a, N - 1);
    }
}
```
- $O(1)$
 - $O(\log N)$
 - $O(N)$
 - $O(N \log N)$
 - $O(N^2)$
 - $O(N^3)$
 - $O(2^N)$
 - $O(N!)$

Correct answer:

$O(N^3)$

ComplexCode24

What is the asymptotic complexity (big-O) of the following code?

```
public int f(int N) {  
    if (N <= 0) {  
        return 0;  
    } else if (N % 2 == 0) {  
        return 1 + f(N - 1);  
    } else {  
        return 1 + f(N / 2);  
    }  
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(\log N)$

ComplexCode25

What is the asymptotic complexity (big-O) of the following code?

```
public int f3(int N) {  
    if (N <= 0) {  
        return 0;  
    } else if (N % 2 == 0) {  
        return 1 + f(N - 1);  
    } else {  
        return 1 + f(N - 2);  
    }  
}
```

- $O(1)$
- $O(\log N)$
- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N^3)$
- $O(2^N)$
- $O(N!)$

Correct answer:

$O(N)$