

## $O(n)$ vs $O(n \log n)$

### Introduction

When there are two nested loops, one of which is linear and the other logarithmic, you can sometimes find the asymptotic complexity by a “brute force” approach.

### Linear and Logarithmic Time

The following two code fragments take linear time, that is,  $\Theta(n)$ .

1. 

```
for(int i = 0; i < n; i++)
    cout << "Hello!" << endl;
```
2. 

```
for(int i = n; i > 0; i--)
    cout << "Hello!" << endl;
```

The following two code fragments take logarithmic time, that is,  $\Theta(\log n)$ .

3. 

```
for(int i = 1; i < n; i = 2*i)
    cout << "Hello!" << endl;
```
4. 

```
for(int i = n; i > 0; i = i/2)
    cout << "Hello!" << endl;
```

### 0.1 Nested Linear and Logarithmic Loops

For each of these examples, we give brute force analysis, where  $n = 9$ .

5. 

```
for(int i = 1; i < n; i++)
    for(int j = 1; i < n; j=2*j)
        cout << "Hello!" << endl;
```

```

i  j
1  1 2 4 8
2  1 2 4 8
3  1 2 4 8
4  1 2 4 8
5  1 2 4 8
6  1 2 4 8
7  1 2 4 8
8  1 2 4 8
```

The answer is  $\Theta(n \log n)$ .

```
6. for(int i = 1; i < n; i++)
    for(int j = i; i < n; j=2*j)
        cout << "Hello!" << endl;
```

```

i   j
1   1 2 4 8
2   2 4 8
3   3 6
4   4 8
5   5
6   6
7   7
8   8
```

The answer is  $\Theta(n)$ .

```
7. for(int i = 1; i < n; i++)
    for(int j = 1; j >= i; j=2*j)
        cout << "Hello!" << endl;
```

```

i   j
1   1
2   1 2
3   1 2
4   1 2 4
5   1 2 4
6   1 2 4
7   1 2 4
8   1 2 4 8
```

The answer is  $\Theta(n \log n)$ .

```
8. for(int i = 1; i < n; i=2*i)
    for(int j = 1; j < n; j++)
        cout << "Hello!" << endl;
```

```

i   j
1   1 2 3 4 5 6 7 8
2   1 2 3 4 5 6 7 8
4   1 2 3 4 5 6 7 8
8   1 2 3 4 5 6 7 8
```

The answer is  $\Theta(n \log n)$ .

```
9. for(int i = 1; i < n; i=2*i)
    for(int j = 1; j <= i; j++)
        cout << "Hello!" << endl;
```

```
  i   j
  1   1
  2   1 2
  4   1 2 3 4
  8   1 2 3 4 5 6 7 8
```

The answer is  $\Theta(n)$ .

```
10. for(int i = 1; i < n; i=2*i)
     for(int j = i; j < n; j++)
         cout << "Hello!" << endl;
```

```
  i   j
  1   1 2 3 4 5 6 7 8
  2   2 3 4 5 6 8 8
  4   4 5 6 7 8
  8   8
```

The answer is  $\Theta(n \log n)$ .