

# Complexity of an Algorithm

By

Nuttachot Promrit

char h = 'y'; // This will be executed 1 time

int abc = 0; // This will be executed 1 time

```
for (int i = 0; i < N; i++) {  
    cout << "Hello World";  
}
```

int i = 0  $\rightarrow$  1

i < N  $\rightarrow$  N + 1

i++  $\rightarrow$  N

cout << "Hello World";  $\rightarrow$  N

$$1 + N + 1 + N + N = 3N + 2$$

```
for ( i = 0; i < N; i++ )  
    statement;
```

N

```
for ( i = 0; i < N; i++ ) {  
    for ( j = 0; j < N; j++ )  
        statement;  
}
```

$N * N$

```
while ( low <= high ) {  
    mid = ( low + high ) / 2;  
    if ( target < list[mid] )  
        high = mid - 1;  
    else if ( target > list[mid] )  
        low = mid + 1;  
    else break;  
}
```

1 100000

2 50000

3 25000

4 12,500

5 6,250

6 3,125

7 1,562

8 781

9 390

10 195

11 97

12 48

13 24

14 12

15 6

16 3

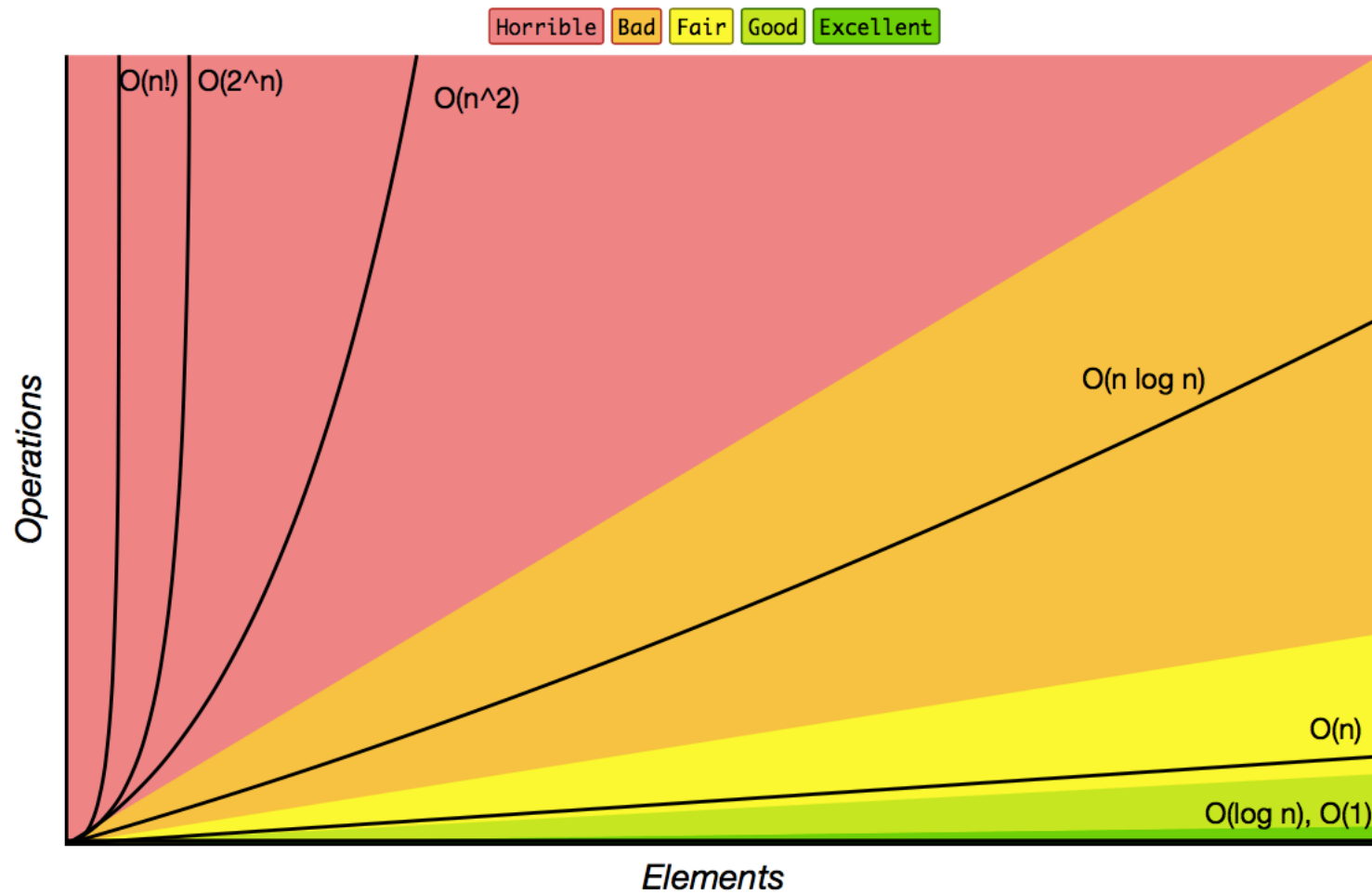
17 1

2 4 8 16 32 64 128 256 512 1024 2048 4096 8,192 16,384 32,768 65,536 131,072

```
while ( low <= high ) {  
    mid = ( low + high ) / 2;  
    if ( target < list[mid] )  
        high = mid - 1;  
    else if ( target > list[mid] )  
        low = mid + 1;  
    else break;  
}
```

Log (N)

## Big-O Complexity Chart



Pic from <http://bigocheatsheet.com>