

$\log_{10}(n) < \ln(n)$
 $\frac{\log_{10}(n)}{\ln(n)} \rightarrow o(\log(n))$
 $\log_e(10) \approx 2.30$

↓

$O(\log_{10}(n))$ $O(\ln(n))$ $O(\sqrt{n})$ $O(n \log n)$ $O(n^2)$ $O(n^4)$

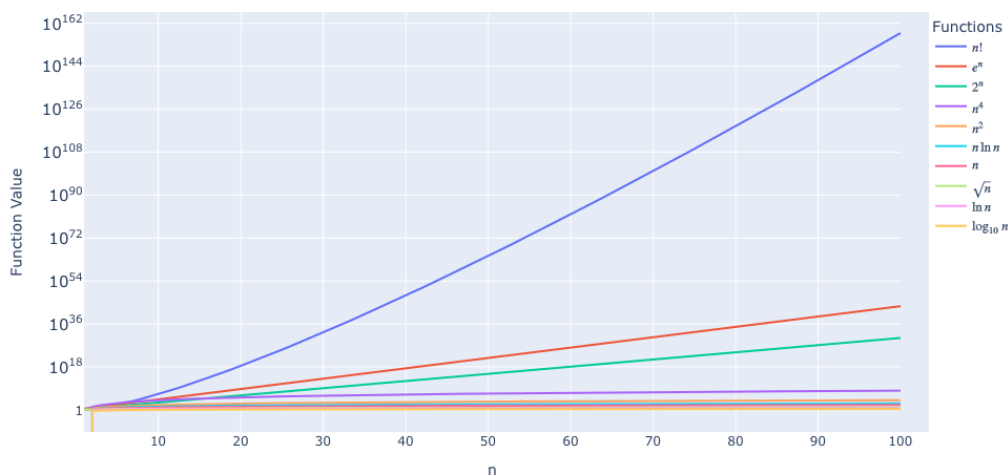
$O(n)$

$$2 < e \Rightarrow 2^n < e^n$$

$$\left. \begin{aligned} n! &= 1 \times 2 \times \dots \times n \\ e^n &= e \times e \times \dots \times e \end{aligned} \right\} \downarrow 2^n < e^n < n!$$

<https://eric-han.com/teaching/AY2425S1/CS1010/P20.1.html>

Comparison of Various Mathematical Functions



What is the Big-O running time of the following code, in terms of n ?

a) $O(n \times n/2) \sim O(n^2)$

```

1  for (long i = 0; i < n; i += 1) {
2      for (long j = 0; j < n; j += 2) {
3          cs1010_println_long(i + j);
4      }
5  }

```

Handwritten annotations: $0, 1, 2, \dots, n-1$ for i ; $0, 2, \dots, n$ for j ; $\lceil n/2 \rceil$ for the number of iterations of j .

b) $O(\log(n) \times \log(n)) \sim O(\log^2 n)$

```

1  for (long i = 1; i < n; i *= 2) {
2      for (long j = 1; j < n; j *= 2) {
3          cs1010_println_long(i + j);
4      }
5  }

```

Handwritten annotations: $1, 2, 4, \dots$ for i ; $1, 2, 4, \dots$ for j ; $\lceil \log_2(n) \rceil$ for the number of iterations of j .

c) $O(1 + 2 + \dots + n) = O\left(\sum_{i=1}^n i\right) = \frac{n(n+1)}{2} = O(n^2)$ *Arith Series*

```

1  for (long j = 0; j < n; j += 1) {
2      for (long i = 0; i < j; i += 1) {
3          cs1010_println_long(i + j);
4      }
5  }

```

Handwritten annotations: $0, 1, 2, \dots, n-1$ for j ; $0, \dots, n-1$ for i ; $0, 1, 2$ for i when $j=2$; $0, 1$ for i when $j=1$; 0 for i when $j=0$.

d) $O(2^1 + 2^2 + \dots + 2^n) = O\left(\sum_{k=1}^n 2^k\right) = 2(2^n - 1) = O(2^n)$ *Geo. Series*

```

1  long k = 1;
2  for (long j = 0; j < n; j += 1) {
3      k *= 2;
4      for (long i = 0; i < k; i += 1) {
5          cs1010_println_long(i + j);
6      }
7  }

```

Handwritten annotations: $0, 1, 2, \dots, n-1$ for j ; 2^n for the number of iterations of i when $j=n-1$; $k=4$ for $j=2$; $0, 1, 2, 3$ for i when $k=4$; $4 = 2^2$; $k=2$ for $j=1$; $0, 1$ for i when $k=2$; $2 = 2^1$.

a) Express the running time of the following function as a recurrence relation: *Laetis*

```
1 void foo(long n) {  
2     if (n == 1) {  
3         return 1;  
4     }  
5     return foo(n/2) + 2;  
6 }
```

$$foo(n) = \begin{cases} 1 & \text{if } n = 1 \\ \frac{foo(n/2) + 2}{1} & \text{else} \end{cases}$$

What is its running time?

$$T(n) = T(n/2) + 1, T(1) = 1 \quad \text{1 action}$$

$$T(8)$$

$$T(4) + 1 \quad \log_2(n)$$

$$T(2) + 1$$

$$T(1) + 1$$

$$1$$

$$T(8) = 4$$

$$T(n) = \log_2(n) + 1$$

$$O(\log n)$$

b) Express the running time of the following function as a recurrence relation:

```

1 void foo(long n) {
2     if (n == 1) {
3         return 1;
4     }
5     for (long i = 0; i < n; i += 1) {
6         cs1010_println_long(i);
7     }
8     return foo(n - 1);
9 }

```

$$T(n) = T(n-1) + n$$

What is its running time?

$$T(1) = 1$$

$$\begin{aligned}
 &T(8) \\
 &\quad T(7) + 8 \\
 &\quad\quad T(6) + 7 \\
 &\quad\quad\quad T(5) + 6 \\
 &\quad\quad\quad\quad \dots \\
 &\quad\quad\quad\quad\quad T(1) + 2 \\
 &\quad\quad\quad\quad\quad\quad | \\
 &\quad\quad\quad\quad\quad\quad |
 \end{aligned}$$

$$T(8) = \sum_{i=1}^8 i$$

$$T(n) = \frac{n(n-1)}{2}$$

$$T(n) = O(n^2)$$