$\log_{10}(n) < \ln(n) \rightarrow O(\log(n))$

$\frac{\ln(n)}{\log_e(10)}$ ~ 2.30

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

$O(n)$

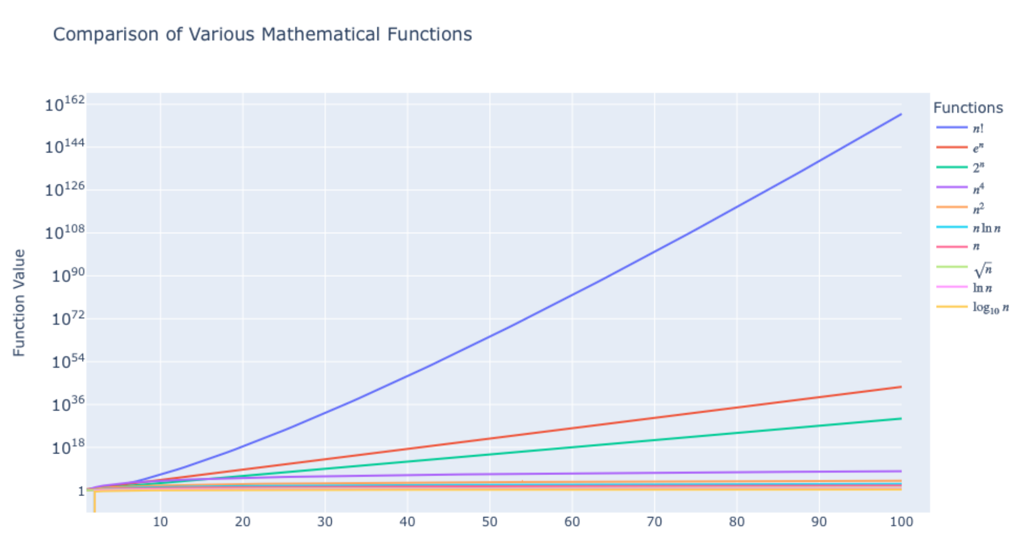
$2 < e \Rightarrow 2^n < e^n$

$n! = 1 \times 2 \times \dots \times n$

$e^n = e \times e \times \dots \times e$

$< 2^n < e^n < n!$

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



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

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

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

$0, 1, 2 \dots n-1$ $\quad n$

$0, 2, \dots$ $\quad \lceil n/2 \rceil$

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

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

$1, 2, 4, \dots$ $\quad \lceil \log_2(n) \rceil$

$1, 2, 4, \dots$ $\quad \lceil \log_2(n) \rceil$

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

```
for (long j = 0; j < n; j += 1) {
  for (long i = 0; i < j; i += 1) {
    cs1010_println_long(i + j);
  }
}
```

$0, 1, 2 \dots n-1$ $\quad n$

$0, \dots n-1$ $\quad n$

$\dots$

$0, 1$ $\quad 2$

$0,$ $\quad 1$

$\cancel{0}$ $\quad 0$

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

```
long k = 1;
for (long j = 0; j < n; j += 1) {
  k *= 2;
  for (long i = 0; i < k; i += 1) {
    cs1010_println_long(i + j);
  }
}
```

$k=1$

$0, 1, 2 \dots n-1$ $\quad n$

2^n

$k=4$ $\quad 0, 1, 2, 3$ $\quad 4 = 2^2$

$k=2$ $\quad 0, 1$ $\quad 2 = 2^1$

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

1 action

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

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

1 action

What is its running time?

$T(n) = T(n/2) + 1$, $T(1) = 1$

$T(8)$

$T(4) + 1$

$\log_2(n)$

$T(2) + 1$

$T(1) + 1$

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:

```
void foo(long n) {
  if (n == 1) {
    return 1;
  }
  for (long i = 0; i < n; i += 1) {
    cs1010_println_long(i);
  }
  return foo(n - 1);
}
```

$0, 1, \ldots n-1 \longrightarrow n$

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

What is its running time?

$$T(1) = 1$$

$T(8)$

$T(7) + 8$

$T(6) + 7$

$T(5) + 6$

$\ldots$

$T(1) + 2$

1

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

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

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