$O(g(n))$ → class of functions that grows asymptotically slower than $g(n)$

measured func ← $f(n) = O(g(n))$ → bounding func.

means:

$\exists$ some $m > 0$, n_0 : starting value of n

where $|f(n)| \leq m g(n)$ $\forall$ $n > n_0$

$\forall$ for ALL

Example: Ins. sort

```
Insertion Sort
Version: 2
Input: array
Output: None, sort in place
Steps:
1. n = length of array
2. For outer_index in Range(from 1 to n-1), do:
    2.1 inner_index = outer_index  # start with the i-th element
    2.2 temporary = array[inner_index]
    2.3 As long as (inner_index > 0) AND (temporary < array[inner_index - 1]), do:
        2.3.1 array[inner_index] = array[inner_index - 1]) # shift to the right
        2.3.2 inner_index = inner_index - 1  # move to the left
    2.4 array[inner_index] = temporary # save temporary to its final position
```

n-1 times

assume: i seconds per instruction

$-i$ $-i$ $-i$ $-i$ $-i$ $-i$

→ worst case: outer_index times

Time to execute: $2i$ per inner loop, $3i$ per outer loop

$$(1 + 2 + \dots + n-1) * 2i + 3in = \frac{n*(n-1)}{2} * 2i + 3in.$$

$$= \frac{n^2 - n}{2} * 2i + 3in$$

guess: insertion sort = $O(n^2)$.

$$\frac{n^2-n}{2} * 2i + 3in = O(n^2)$$

Starry number of elements in array

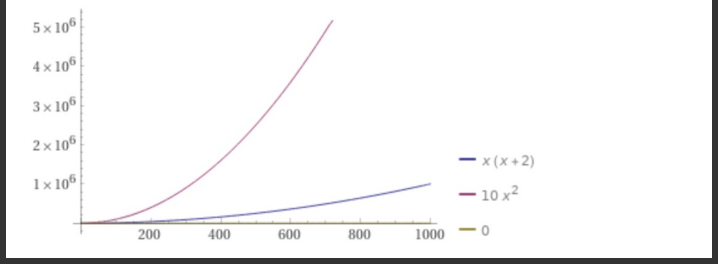
can we find some $m > 0$, and $n_0 > 0$, such that:

$$\left|\frac{n^2-n}{2} 2i + 3in\right| \leq m|n^2| \quad \forall \quad n > n_0 ?$$

assume $i = 1$,

plot $y = \left(\frac{1}{2}(x^2 - x)\right) \times 2 + 3x$, $y = 10x^2$, $y = 0$ for $x = 0$ to 1000

$m = 10$ certainly works.



if you assume $i = k$, where k is a constant, there is always some other m whereby the above condition is true.

$\therefore$ insertion sort = $O(n^2)$

$\Omega(g(n))$: class of functions that grows asymptotically faster than $g(n)$.

$f(n) = \Omega(g(n))$ means:

there exists some $m > 0$, n_0, that

$$|f(n)| \geq m|g(n)| \quad \forall \; n > n_0$$

$\Theta(g(n))$: class of functions that grow asymptotically similar to $g(n)$.

$f(n) = \Theta(g(n))$ means:

there exist some $m_1, m_2 > 0$, n_0 where:

$$|f(n)| \geq m_1 |g(n)| \quad \forall n \geq n_0$$
$$|f(n)| \leq m_2 |g(n)|$$

Usefulness:

1. $O(g(n))$: worst case complexity
2. $\Omega(g(n))$: Best case complexity

} to find average in practice.

3. $\Theta(g(n))$: Both best and worst case complexity

to proof that $f(n) = \Theta(g(n))$: show that

$f(n) = O(g(n))$

$f(n) = \Omega(g(n))$ or $g(n) = O(f(n))$

Eg: $f(n) = 3n^2$

is $f(n) = \theta(n^2)$?

① $3n^2 \leq m_1 n^2$, $m_1 \geq 3$ $\forall$ $n > n_0$.

② $3n^2 \geq m_2 n^2$, $m_2 \leq 3$ $\forall$ $n > n_0$

$\therefore$ $f(n) = \theta(n^2)$.

Simplifying Insertion sort:

$\frac{n^2 - n}{2} * 2i + 3n$ ($O(n^2)$, $O(n)$, $O(n)$)

$= \frac{O(n^2) - O(n)}{2} * O(1) + O(n)$ (2 → constant doesn't affect asymptotic complexity.)

$O(1)$ constant, does not depend on "n"

$= (O(n^2) - O(n))\, O(1) + O(n)$

$= \boxed{O(n^2)}$. dominating term.

Factorial Stirling formula: $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n = O(n^n)$.

summary: $O(n^n) > O(e^n) > O(n^i)$ ($i \geq 1$) $> O(n \log n) > O(n) > O(\log n) > O(1)$