Asymptotic Complexity Zoo

largest n we can handle in 1 sec (10^8 ops/sec).

$\Theta(1)$ (constant)

i.e. does not depend on size of input.

Examples:

- appending to a list.
- indexing into array
- push/pop stack or queue
- add/remove from either end of a double-ended queue (deque)
- arithmetic
- hash table ops. (set or dict. in Python eg)

$\Theta(\lg n)$ — logarithmic.

$\lg n \le 10^8$
$\hookrightarrow n \le 2^{10^8}$

$\lg n$ means $\log_2$.

Thm $\Theta(\log_a n) = \Theta(\log_b n)$ for all $a, b > 1$.

Examples of things that are $\Theta(\lg n)$:

- Height of balanced binary tree of size n.
- Binary search
- Heap insertion
- # of bits needed to represent n.

Thm $\lg n$ is $o(n^x)$ for all $x > 0$.

$\Theta(\sqrt{n})$

$n \le 10^{16}$ in one second.

We won't see this much, but it is possible.

$\Theta(n)$ — linear

$n \le 10^8$ in one second.

Examples:

- max/sum/min etc. of unsorted list.
- linear search

- adding/removing from beginning of a list.
- adding to the end of a string

$\Theta(n \lg n)$ - "linearithmic" $n \leq 5 \cdot 10^6$??

- Merge sort, quick sort., other divide + conquer.

$\Theta(n^2)$ - quadratic $n \leq 10^4$ in 1 second.

Examples:

- 2 nested loops (* usually)
- insertion sort / bubble sort
- $1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} = \Theta(n^2)$
- Building string of length n by repeated append. $(1 + 2 + 3 + \dots + n)$.
- # of ways to choose 2 things out of n. $\binom{n}{2}$.

$\Theta(2^n)$ - exponential. $n \leq 25$

- # nodes in tree of ht n.
- $1 + 2 + 4 + 8 + \dots + 2^n = 2^{n+1} - 1 = \Theta(2^n)$.