

Asymptotic Complexity Zoo

Biggest n we can process in 1 second.
 ∞

Constant Time: $\Theta(1)$

Things that don't depend on size of input.

- Checking even/odd
- Accessing array index
- push/pop from stack
- insert/lookup in hash table

Logarithmic time: $\Theta(\log n)$

unimaginably large.

Repeatedly halving. Examples:

- Lookup in a binary search tree
- # of bits to represent n.
- binary search.

Note: $\log_a n = \frac{\log_c n}{\log_c a} = \frac{1}{\log_c a} \cdot \log_c n$.

\uparrow constant.

Hence base of log does not matter for big-O, etc.

$$\lg = \log_2.$$

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

Proof. $\lim_{n \rightarrow \infty} \frac{\ln n}{n^x} = \lim_{n \rightarrow \infty} \frac{1/n}{x n^{x-1}} = \lim_{n \rightarrow \infty} \frac{1}{x n^x} = 0$.

Linear time: $\Theta(n)$.

(n = 100 million)
in 1 sec

Examples: ^{or reasoning}

- Inserting @ start of an array.
- Linear search.
- Finding max/min/sum of a list.
- Merging 2 sorted lists

Linearithmic $\Theta(n \lg n)$

$n = 4.5$ million.

Examples:

- Merge sort, Quick sort

Quadratic $\Theta(n^2)$

$n = 10$ thousand.

Examples:

- Insertion sort, bubble sort, gnome sort
- Repeated string append
- 2 nested loops (often)
- $1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$ is $\Theta(n^2)$
- # of pairs of n things.

Polynomial time $\Theta(n^k)$

- k nested loops
- Number of size- k subsets of n things
ie. $\binom{n}{k}$ is $\Theta(n^k)$

Thm: n^j is $o(n^k)$ when $j < k$.

Exponential time $\Theta(2^n)$.

$n = 26$ in 1 second.

Examples:

- # of numbers w/ n bits.
- # of subsets of n things.
- Size of a depth- n tree
- $1 + 2 + 4 + 8 + 16 + \dots + 2^n = 2^{n+1} - 1$ is $\Theta(2^n)$.

Thm n^k is $o(r^n)$ for all $k \geq 0$ and $r > 1$.

