When proving which one is faster, we can use L'Hospital's rule.

通过左右移项，或者左右相乘, multiply / divide

Exponentials grows much more quickly than polnomials, a^n >>> n^x

exponentials > polynomial

logarithmic < polynumial

Proofing methods

1. Counter-Example

show an example which does not fit with the theorem

draw the conclusion that the theorem is incorrect

2. Contradiction

assume the opposite of the theorem

derive a contradiction

draw the conclusion that the theorem is correct.

3. Induction

Prove for one or more base case (e.g. n = 1)

assume for one or more anonymous values (e.g. k)

prove for the next value it is true (e.g. k + 1)

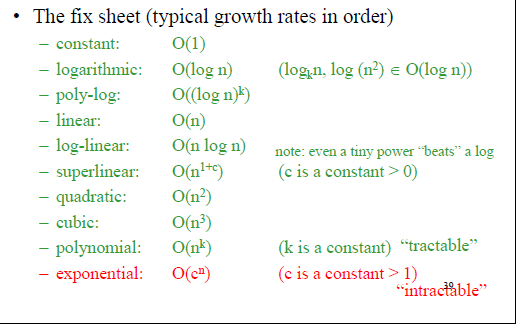
dwas the conclusion that the theorem is correct.

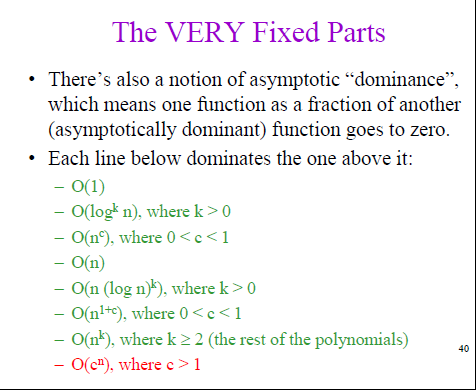
Asymptotic Analysis->Eliminate low order terms

->Eliminate coefficients

nlogn is in O(n^2)

Typical growth rates in ascending order





The way to analyse

Normal operations->constant time

Consecutive statements->sum of times

Conditionals->sum of branches, condition

Loops->Sum of iterations

Function calls->Cost of function body

Constant time O(1)-> This means that the algorithm requires the same fixed number of steps regardless of the size of the task.

Example: Push and Pop operations for a stack (containing n elements); Insert and Remove operations for a queue.

Linear time O(n)-> This means that the algorithm requires a number of steps proportional to the size of the task.

Example: Traversal of a list (a linked list or an array) with n elements;

Finding the maximum or minimum element in a list, or sequential search in an unsorted list of n elements;

Traversal of a tree with n nodes;

Quadratic time O(n^2)->The number of operations is proportional to the size of the task squared

Logarithmic time O(logn)

Example: Binary search in a sorted list of n elements; Insert and Find operations for a binary search tree with n nodes(logn levels); Insert and Remove operations for a heap with n nodes

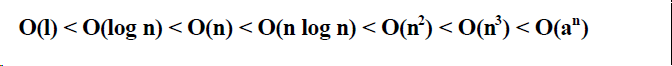
O(n\*logn)

Example: quicksort, mergesort

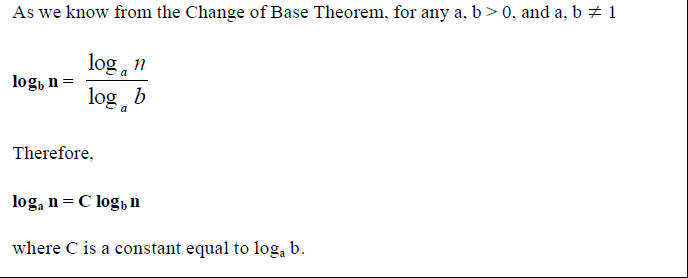
Exponential time O(a^n), a > 1

Exponential > Polynomial (linear, quadratic, cubic, ect.)

ORDER:



Change of Base Theorem:



Since functions that differ only by a constant factor have the same order of growth, **O(log2 n)** is the same as **O(log n).** Therefore, when we talk about logarithmic growth, the base of the logarithm is not important, and we can saysimply **O(log n).**