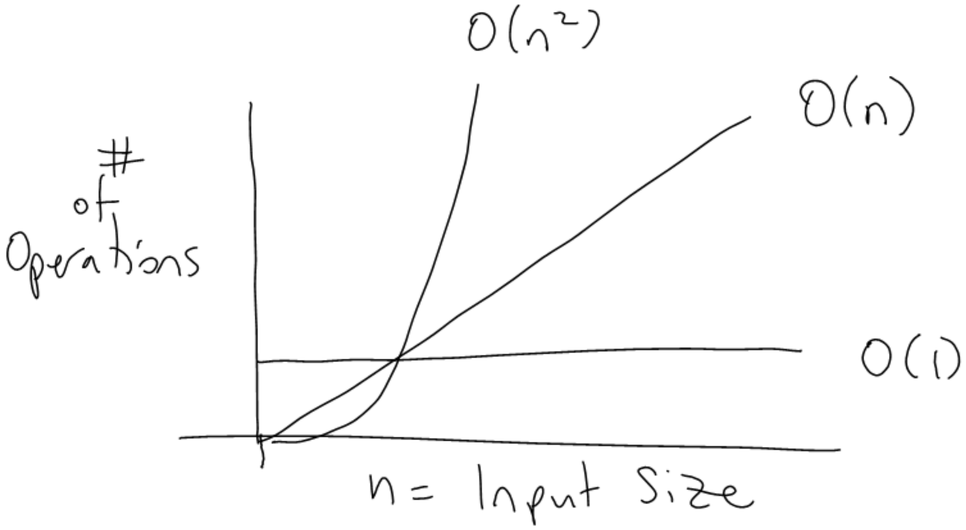
4.1 Algorithmic Complexity Cont'd

Sorting Integers

Input: 11, 7, 6, 21

Output: 6, 7, 11, 21

Harder as # of integers increases.



An algorithm that is $O(1)$ performs faster than $O(n)$, $O(n^2)$ for large n.

An algorithm that is $O(n^2)$ performs slower than $O(1)$, $O(n)$ for large n.

4.2 Factorial and Exponential Growth

Recall: Factorials (Section 3.1)

$$3! = 3 \times 2 \times 1 = 6$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

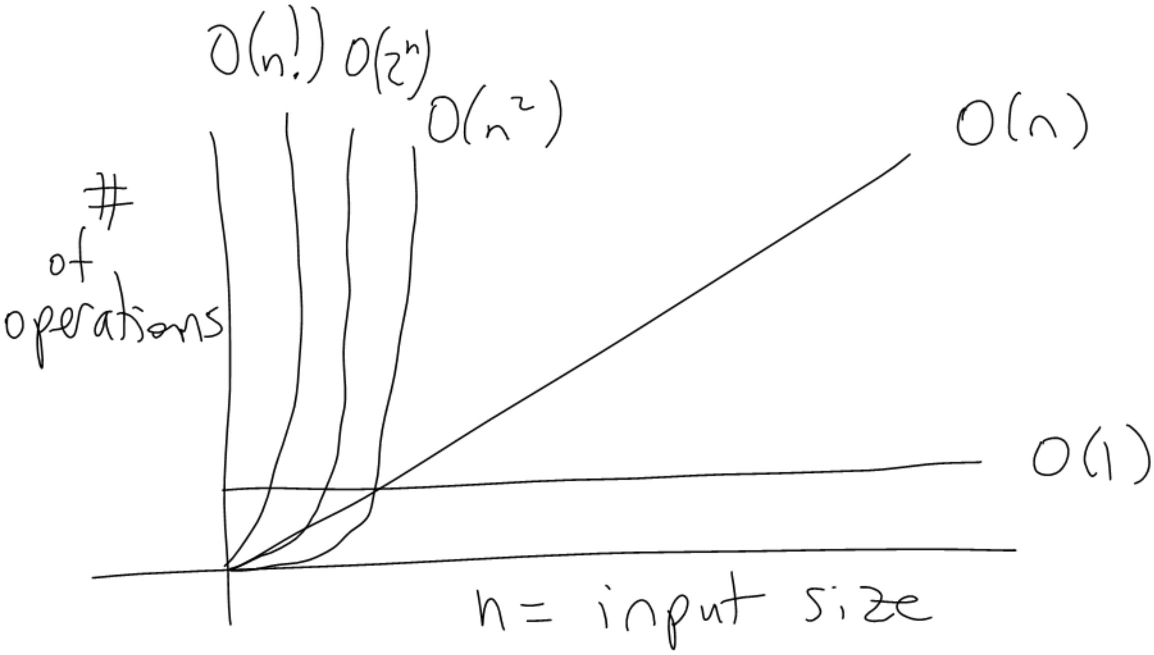
$$n! = n \times (n-1) \times (n-2) \times \ldots \times 1$$

n	polynomial n^2	exponential 2^n	factorial $n!$
1	1	2	1
2	4	4	2
3	9	8	6
4	16	16	24
5	25	32	120
⋮			
50	2500	1.1×10^{15}	3.0×10^{64}

Fact

2^n is dominant over $n^2, n^3, n^{1000}, \ldots$

$n!$ is dominant over $2^n, n^2, n^3, n^4, \ldots$



Ex: Find the order of the expression

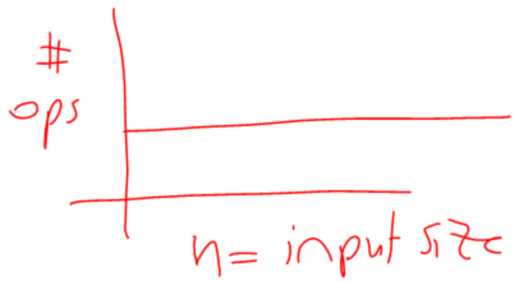
a) $9n^2 + 8(2^n)$

$O(2^n)$

b) $4!$

$= 24$

$O(1)$



c) $8 \cdot n!$

$O(n!)$

d) $20n(n+2)$

$= 20n^2 + 40n$

$O(n^2)$

e) $9n^2 + 3n! + 8(2^n)$

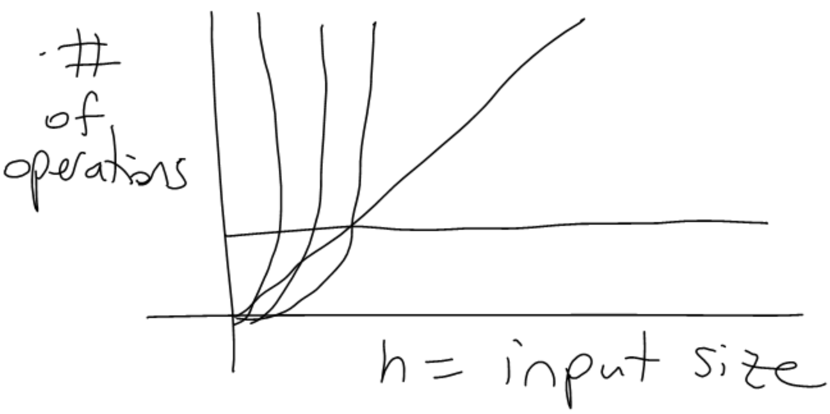
$O(n!)$

Ex: Rank the following from smallest to largest:

$O(n)$, $O(2^n)$, $O(n!)$, $O(1)$, $O(n^2)$

$O(1)$, $O(n)$, $O(n^2)$, $O(2^n)$, $O(n!)$

Ex: Label the curves



4.3 Logarithmic Growth

$2^{?} = 16$ $\qquad$ $2^{4} = 16$ $\qquad$ $\log_2 16 = 4$

$2^{?} = 2$ $\qquad$ $2^{1} = 2$ $\qquad$ $\log_2 2 = 1$

$\log_2 1 = 0$ $\qquad$ $2^{?} = 1$

$\log_{10} 100 = 2$ $\qquad$ $10^{?} = 100$

$\log_{10} 0.1 = -1$ $\qquad$ $10^{?} = 0.1$

$\log_2 32 = 5$ $\qquad$ $2^{?} = 32$

Notation: $\log n$ means $\log_{10} n$

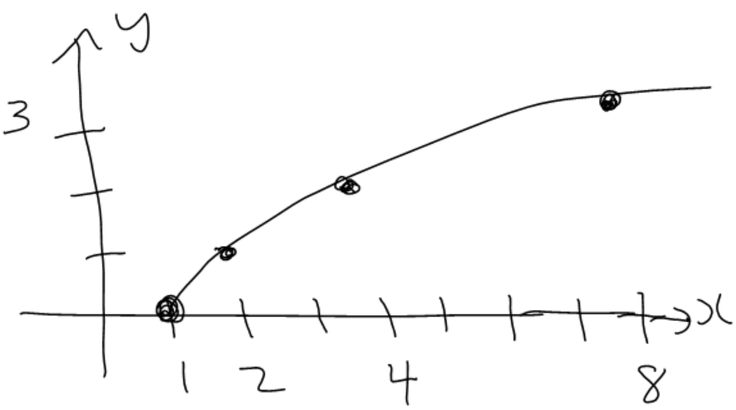
$\log 100 = \log_{10} 100 = 2$ $\qquad$ $10^{?} = 100$

$\log 0.01 = \log_{10} 0.01 = -2$ $\qquad$ $10^{?} = 0.01$

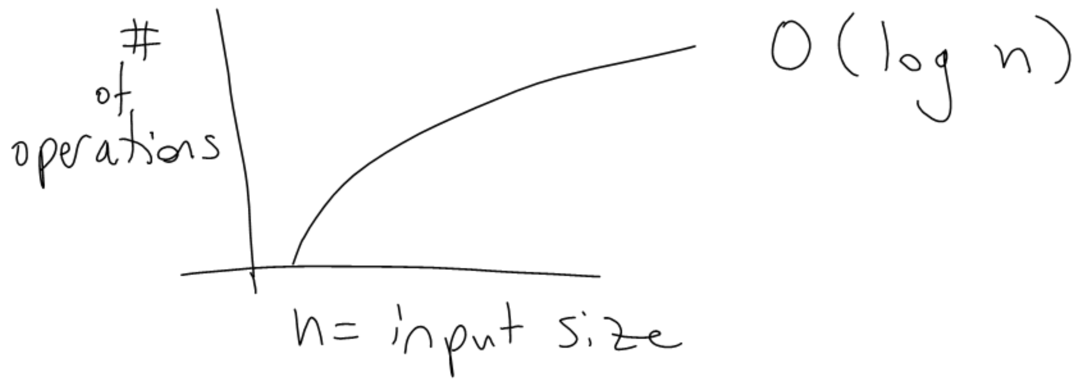
On Calculator

$$\log_2 128 = \frac{\log 128}{\log 2} = 7$$

x	$y = \log_2 x$
1	0
2	1
4	2
8	3



The graph is always increasing.
There is no maximum value.



Ordered List: 2, 5, 8, 9, 12, 17, 23, 40

In which position is 23?

Bad Algorithm

Check 1st position

Check 2nd position

⋮

Until desired # is found

$\boxed{\text{This is } O(n)}$

Good Algorithm

1. Break list in 2 halves
2. Select lower half or upper half, and record positions.
3. Repeat Steps 1 and 2 until desired # is found.

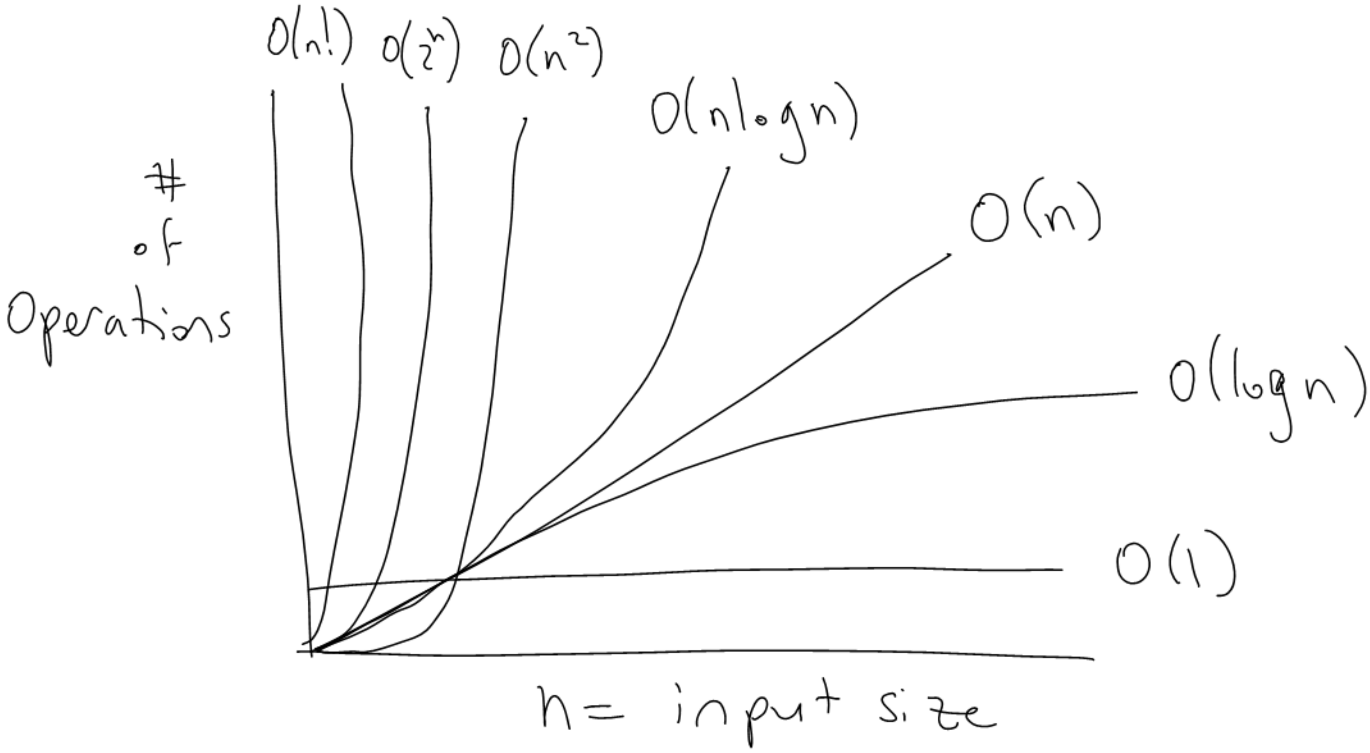
Ex: 2, 5, 8, 9, 12, 17, 23, 40

~~2, 5, 8, 9~~ 12, 17, 23, 40 Positions 5-8

~~12, 17~~ 23, 40 Positions 7-8

23 ~~40~~ Position 7

This is $O(\log n)$



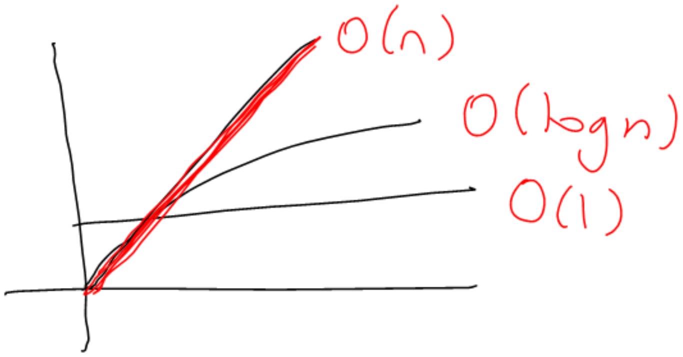
Note: Curves may cross when n is small.

Ex: Order these from smallest to largest:

$O(n^2)$, $O(n\log n)$, $O(2^n)$, $O(n!)$, $O(1)$, $O(n)$, $O(\log n)$

$O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$, $O(n!)$

Ex: Label the curves



Ex: Find the order:

a) $3n + 4\log n$

$O(n)$

b) $3n + 4n\log n$

$O(n \log n)$

c) $(\log n)(3 + 2n)$

$= 3\log n + 2n\log n$

$O(n \log n)$

d) $3 + 4 \log n$

$O(\log n)$

$\log_{10} 1000 = 3$ $\qquad$ $10^? = 1000$

$\log_2 16 = 4$ $\qquad$ $2^? = 16$