# Algorithms: Review

Iyad Kanj

DePaul University

# Logarithms

Let $a, b > 1$ and $x, y > 0$

$\log_a x = z \iff a^z = x$
Example: $\log_2 32 = 5; \log_3 81 = 4$

We will write lg for $\log_2$

**Properties of logarithms**

- $\log_a a = 1$ and $\log_a 1 = 0$
- $\log_a x^p = p \log_a x$
  Example: $\lg 64 = \lg 2^6 = 6 \lg 2 = 6$
- $\log_a (x \cdot y) = \log_a x + \log_a y$
  Example: $\lg 24 = \lg (2^3 \cdot 3) = 3 \lg 2 + \lg 3 = 3 + \lg 3$
- $\log_a \left(\frac{x}{y}\right) = \log_a x - \log_a y$
  Example: $\lg \left(\frac{1}{1024}\right) = 0 - \lg 2^{10} = -10$

# Logarithms

**Properties of logarithms**

- $\log_b x = \frac{\log_a x}{\log_a b}$
  Note: $\forall a, b$ : $\log_b x$ and $\log_a x$ are within a multiplicative constant from each other
- $\lg_a x$ is an increasing unbounded function of $x$
- $\lg_a x$ grows "very slowly"
- $\lg_a x$ and $a^x$ are inverses of each other: $a^{\lg_a x} = x$ and $\lg_a a^x = x$

# Summations

**Arithmetic Sequence:**
Each term equals the preceding one plus a fixed constant

**Arithmetic Summation:** $a_1 + \cdots + a_n = (a_1 + a_n) \cdot \frac{n}{2}$

Examples:

$1 + 2 + \cdots + 100 = (1 + 100) \cdot \frac{100}{2} = 5050$

$1 + 2 + \cdots + n$=n(n+1)/2

$1 + 3 + 5 + \cdots + 201 = (1 + 201) \cdot 101/2 = 10201$

**Geometric Sequence:**
Each term equals the preceding one multiplied by a fixed constant

**Geometric Summation:** $a_1 + \cdots + a_n = \frac{a_1(1-r^n)}{1-r}$ $(r \neq 1)$

Special Case: $|r| < 1$ $\quad a_1 + a_2 + \cdots = \frac{a_1}{1-r}$

Examples:

$1 + 2 + \cdots + 1024 = 2047$

$1 + \frac{1}{2} + \frac{1}{4} + \cdots = 2$

$n + \frac{n}{2} + \cdots + 1 \leq 2n$

# Run-Time Analysis

**Searching**

Given an array $A[1 \ldots n]$ of numbers and a number *key*, determine if $key \in A$

**Linear Search**

```
for i=1 to n do
    if A(i)=key then
        return (True);
    end
end
return (False);
```

**Input Size:** $n$
**Operation:** Comparison
**Best case:** 1
**Worst Case:** $n$
**Average Case:** $\approx \frac{n}{2}$

# Run-Time Analysis

## Sorting

Given an array $A[1 \dots n]$ of numbers, rearrange the numbers in $A$ so that $A(1) \leq A(2) \leq \cdots \leq A(n)$

## Insertion Sort

```
for i=2 to n do
    key ← A(i);
    j = i − 1;
    while (j > 0) and (key < A(j))
     do
        A(j + 1) ← A(j);
        j ← j − 1;
    end
    A(j + 1) ← key;
end
```

**Input Size:** $n$
**Operation:** Comparison
**Best case:** $n - 1$
**Worst Case:** $1 + 2 + \cdots + n - 1 = \frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$

# Asymptotic Notations

Let $f, g : \mathbb{N} \to \mathbb{R}$ be asymptotically positive functions

**Definition:** $f(n)$ is $\mathcal{O}(g(n))$ if $\exists c > 0$ and $n_0 \in \mathbb{N}$ such that $f(n) \leq cg(n)$ $\forall n \geq n_0$

Examples:
$n \in \mathcal{O}(n^2)$
$n \log n \notin \mathcal{O}(n)$
$2n + 5 \in \mathcal{O}(n)$
$\frac{1}{2}n^2 + 2n + 10 \in \mathcal{O}(n^2)$
$\log^{100} n \in \mathcal{O}(n^{0.0001})$
$n^{100} \in \mathcal{O}(2^n)$

# Asymptotic Notations

**Definition:** $f(n) \in \Omega(g(n))$ if $\exists c > 0$ and $n_0 \in \mathbb{N}$ such that $f(n) \geq cg(n)$ $\forall n \geq n_0$

**Definition:** $f(n) \in \Theta(g(n))$ iff $f(n) \in \Omega(g(n))$ and $f(n) \in \mathcal{O}(g(n))$

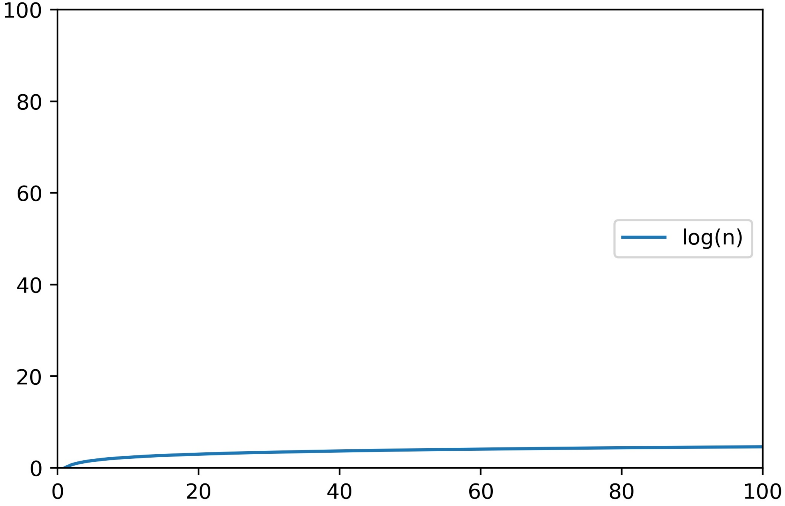
Examples:

$n^{100} + 2n^{90} + n^{70} + N^2 + 1 = \Theta(n^{100})$

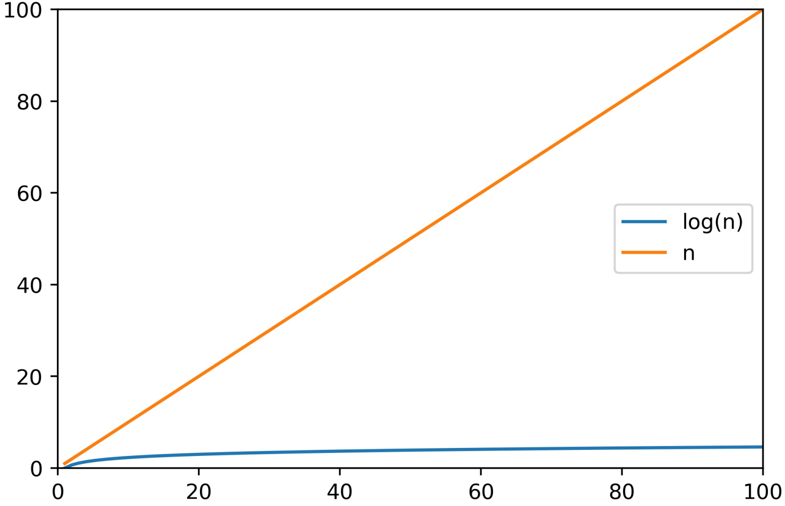
$\log(n!) = \Theta(n \log n)$

**Stirling's Approximation:** $n! \approx \frac{n^n}{e^n}\sqrt{2\pi n}$

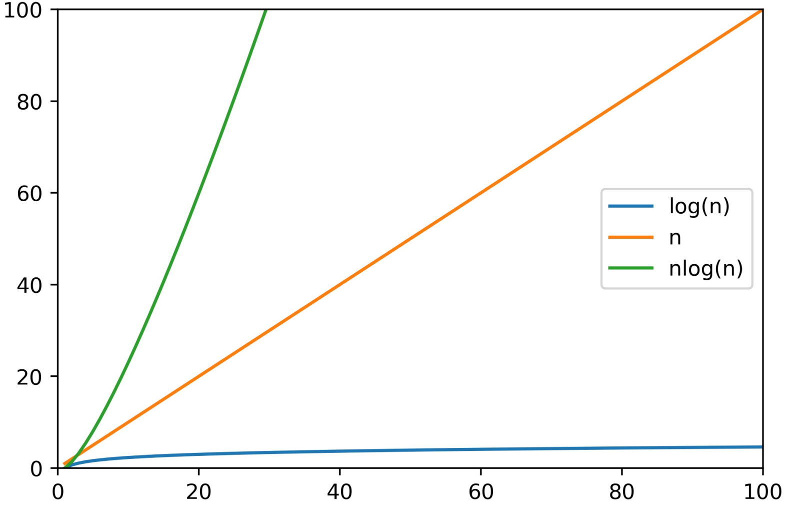
# Asymptotic Growth



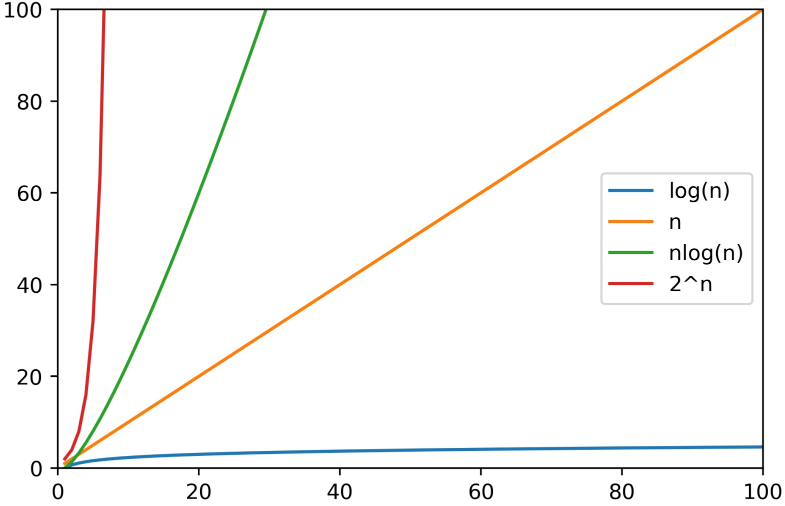
# Asymptotic Growth



# Asymptotic Growth



# Asymptotic Growth



# Asymptotic Growth

**Sort the following functions in a non-decreasing order of their asymptotic growth**

(1) $2n^3 - 5n$
(2) $5n - 3$
(3) $n^n - 2$
(4) $3n^2 - 3n + 1$
(5) $2^n + n + 1$
(6) $4 \lg n - 1$
(7) $n!$
(8) $2n \lg n + 3n$
(9) $10n - 2$
(10) $20$

**Solution: (10), (6), (2)=(9), (8), (4), (1), (5), (7), (3)**