Algorithms

(Algorithm Analysis)

Pramod Ganapathi
Department of Computer Science
State University of New York at Stony Brook

February 7, 2024

Contents

Complexity Analysis

HOME

Complexity analysis

- Framework for analyzing the efficiency/complexity of algorithms
  e.g.: time complexity and space complexity
- Running time of a computer program depends on:
  - Algorithm
  - Input size
  - Input data distribution
  - Machine or computing system
  - Operating system
  - Compiler
  - Programming language
  - Coding
- We analyze the running time of algorithms using
  asymptotic analysis

Complexity analysis

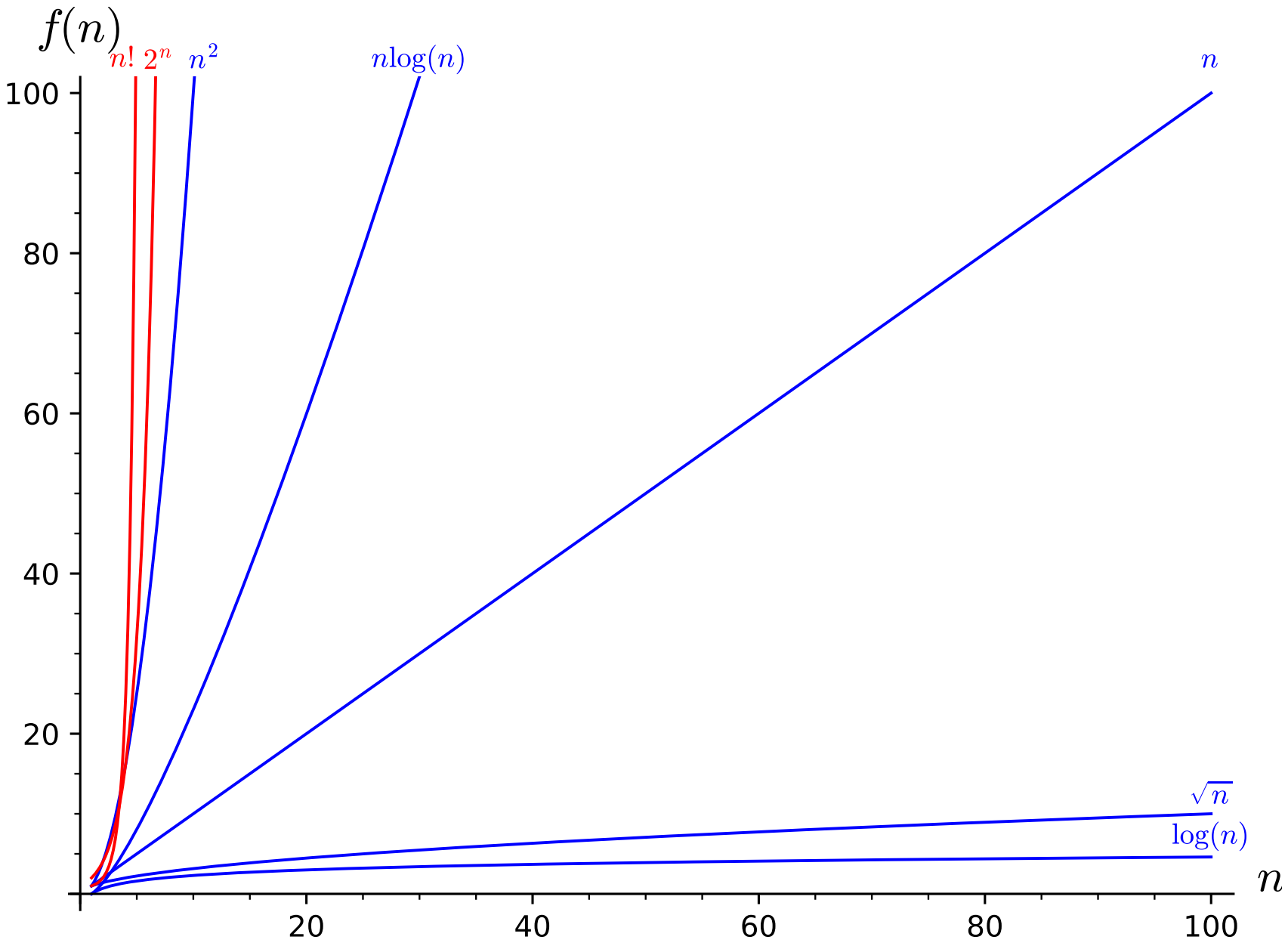
- Time complexity $T(n)$ is $\mathcal{O}\left(n^2\right)$
- Space complexity $S(n)$ is $\Theta\left(n\right)$
- Cache complexity $C(n)$ is $\mathcal{O}\left(\frac{n^2}{BM}\right)$
- Query complexity $Q(n)$ is $\Omega\left(n \log n\right)$ and $\mathcal{O}\left(n^{\log_2 3}\right)$

What do the notations $\Theta, \mathcal{O}, \Omega, o, \omega$ mean?

Complexity analysis

Problem	Running time
Search in a sorted array	$\mathcal{O}(\log n)$
Search in an unsorted array, Integer addition	$\mathcal{O}(n)$
Generate primes	$\mathcal{O}(n \log \log n)$
Sorting, Fast Fourier transform	$\mathcal{O}(n \log n)$
Integer multiplication	$\mathcal{O}(n^2)$
Matrix multiplication	$\mathcal{O}(n^3)$
Linear programming	$\mathcal{O}(n^{3.5})$
Primality test	$\mathcal{O}(n^{10})$
Satisfiability problem	$\mathcal{O}(2^n)$
Traveling salesperson problem	$\mathcal{O}((n-1)!)$
Sudoku, Chess, Checkers, Go	expo. class
Simulate problem, Halting problem	∞
Program correctness, Program equivalence	∞
Integral roots of a polynomial	∞

Polynomial and exponential functions



Units for measuring running time

- Basic operation is the most important operation of the algorithm. Each basic operation takes constant time.
  - Arithmetic operation $(\times, \div, +, -)$
  - Comparison operation $(<, \leq, =, \neq, >, \geq)$
  - Memory operation $(a \leftarrow b, C[i])$
  - Function invocation and return

Units for measuring running time

```
SUM(A[1...n])
sum ← 0
for i ← 1 to n do
|   sum ← sum + A[i]
return sum
```

Runtime is a combination of

- $\approx n$ comparisons
- $\approx n$ additions
- $\approx n$ memory index accesses
- $\approx n$ assignments

Worst-case, best-case, and average-case analysis

- Worst-case complexity $T_{\text{worst}}(n)$ of an algorithm.
  Complexity for the worst-case input of size n for which the algorithm runs the longest among all possible inputs of that size.
- Best-case complexity $T_{\text{best}}(n)$ of an algorithm.
  Complexity for the best-case input of size n for which the algorithm runs the shortest among all possible inputs of that size.
- Average-case complexity $T_{\text{avg}}(n)$ of an algorithm.
  Complexity for a typical or random input of size n.
- Amortized complexity $T_{\text{amortized}}(n)$ of an algorithm.
  Average complexity for a sequence of operations.

Worst-case, best-case, and average-case analysis

Problem

What are the worst-case, best-case, and average-case analyses for the sequential search algorithm?

SEQUENTIAL-SEARCH$(A[1 \ldots n], key)$

Input: An array A and search key key
Output: The index of the first element in A that matches key
or -1 if there are no matching elements

```
i ← 1
while i ≤ n and A[i] ≠ key do
|   i ← i + 1
if i ≤ n then  return i
else  return −1
```

Worst-case, best-case, and average-case analysis

Solution

- $T_{\text{worst}}(n) = n$ ▷ Why?
- $T_{\text{best}}(n) = 1$ ▷ Why?
- $T_{\text{avg}}(n) = \left\{ \begin{array}{ll} \frac{n+1}{2} & \text{if search is successful,} \\ n & \text{if search is unsuccessful.} \end{array} \right\}$ ▷ Why?

Let $p \in [0, 1]$ be the probability of successful search
The prob. of first match occurring at any position be the same
$T_{\text{avg}}(n) = (1 \cdot \frac{p}{n} + 2 \cdot \frac{p}{n} + \cdots + n \cdot \frac{p}{n}) + n \cdot (1 - p)$
Simplifying, $T_{\text{avg}}(n) = \frac{p(n+1)}{2} + n(1 - p)$.

What do you get when you set $p = 1$ or $p = 0$?

Asymptotic Notations

HOME

Asymptotically positive functions

- Let $T(n)$ be asymptotically positive
  i.e., $T(n) : \mathbb{R}^+ \to \mathbb{R}$ such that $\exists n_0 : \forall n \geq n_0, T(n) > 0$
- Number of computations $T(n)$ is asymptotically positive
  Extra space $S(n)$ is asymptotically positive
  Number of cache misses $C(n)$ is asymptotically positive
  Number of queries $Q(n)$ is asymptotically positive

From hereon, we will implicitly assume that

$T(n)$ and other functions are asymptotically positive

Asymptotic notations (using limits)

Definition

- $\boxed{\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0} \Leftrightarrow f(n) \in o\left(g(n)\right)$
- $\boxed{\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty} \Leftrightarrow f(n) \in \omega\left(g(n)\right)$
- Suppose $0 < L < \infty$.

  $\boxed{\lim_{n\to\infty} \frac{f(n)}{g(n)} = L} \Rightarrow f(n) \in \Theta\left(g(n)\right)$

  But $f(n) \in \Theta\left(g(n)\right) \not\Rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)}$ exists

  Example: $f(n) = n^2$ and $g(n) = (2 + \sin n)n^2$.

Asymptotic notations (using limits)

$$\lim_{n\to\infty} \frac{T(n)}{g(n)} = \left\{ \begin{array}{ll} 0 & \text{implies } T(n) \text{ has smaller growth rate than } g(n), \\ c & \text{implies } T(n) \text{ has same growth rate as } g(n), \\ \infty & \text{implies } T(n) \text{ has larger growth rate than } g(n). \end{array} \right\}$$

$$\lim_{n\to\infty} \frac{T(n)}{g(n)} = \left\{ \begin{array}{ll} 0 & \text{implies } T(n) \in o\left(g(n)\right), \\ c & \text{implies } T(n) \in \Theta\left(g(n)\right), \\ \infty & \text{implies } T(n) \in \omega\left(g(n)\right). \end{array} \right\}$$

Asymptotic notations (using sets)

Definition

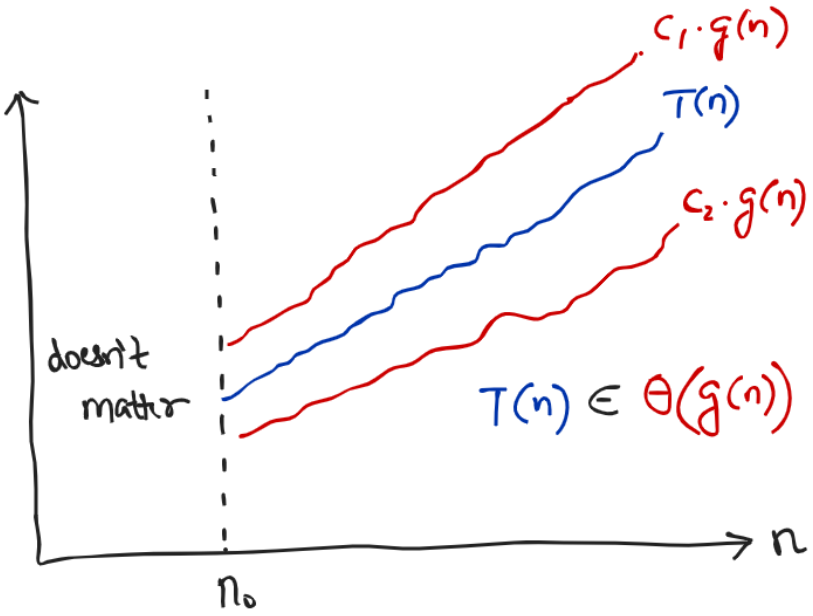
- $\Theta\left(g(n)\right) = \{f(n) \mid \exists c_1, c_2 > 0, \exists n_0 : \forall n \geq n_0,$
  $0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)\}$
- $\mathcal{O}\left(g(n)\right) = \{f(n) \mid \exists c > 0, \exists n_0 : \forall n \geq n_0, 0 \leq f(n) \leq c \cdot g(n)\}$
- $\Omega\left(g(n)\right) = \{f(n) \mid \exists c > 0, \exists n_0 : \forall n \geq n_0, 0 \leq c \cdot g(n) \leq f(n)\}$
- $o\left(g(n)\right) = \{f(n) \mid \boxed{\forall c > 0}, \exists n_0 : \forall n \geq n_0, 0 \leq f(n) < c \cdot g(n)\}$
- $\omega\left(g(n)\right) = \{f(n) \mid \boxed{\forall c > 0}, \exists n_0 : \forall n \geq n_0, 0 \leq c \cdot g(n) < f(n)\}$

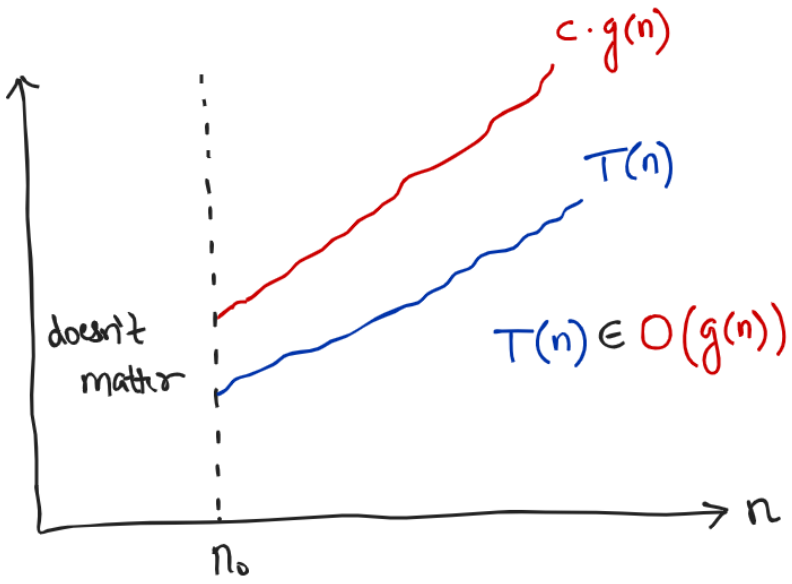
Computing limits with L'Hôpital's rule

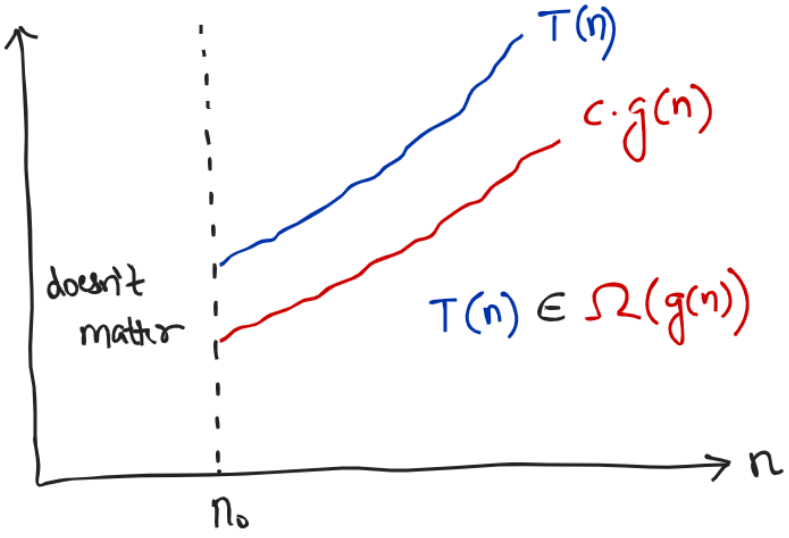
Suppose functions f and g are differentiable on an open interval I except possibly at a point c contained in I.
Suppose $\lim_{x\to c} f(x) = \lim_{x\to c} g(x) = 0$ or $\pm\infty$ and $g'(x) \neq 0$ for all x in I with $x \neq c$, and $\lim_{x\to c} \frac{f'(x)}{g'(x)}$ exists. Then

$$\boxed{\lim_{x\to c} \frac{f(x)}{g(x)} = \lim_{x\to c} \frac{f'(x)}{g'(x)}}$$

Asymptotic notations







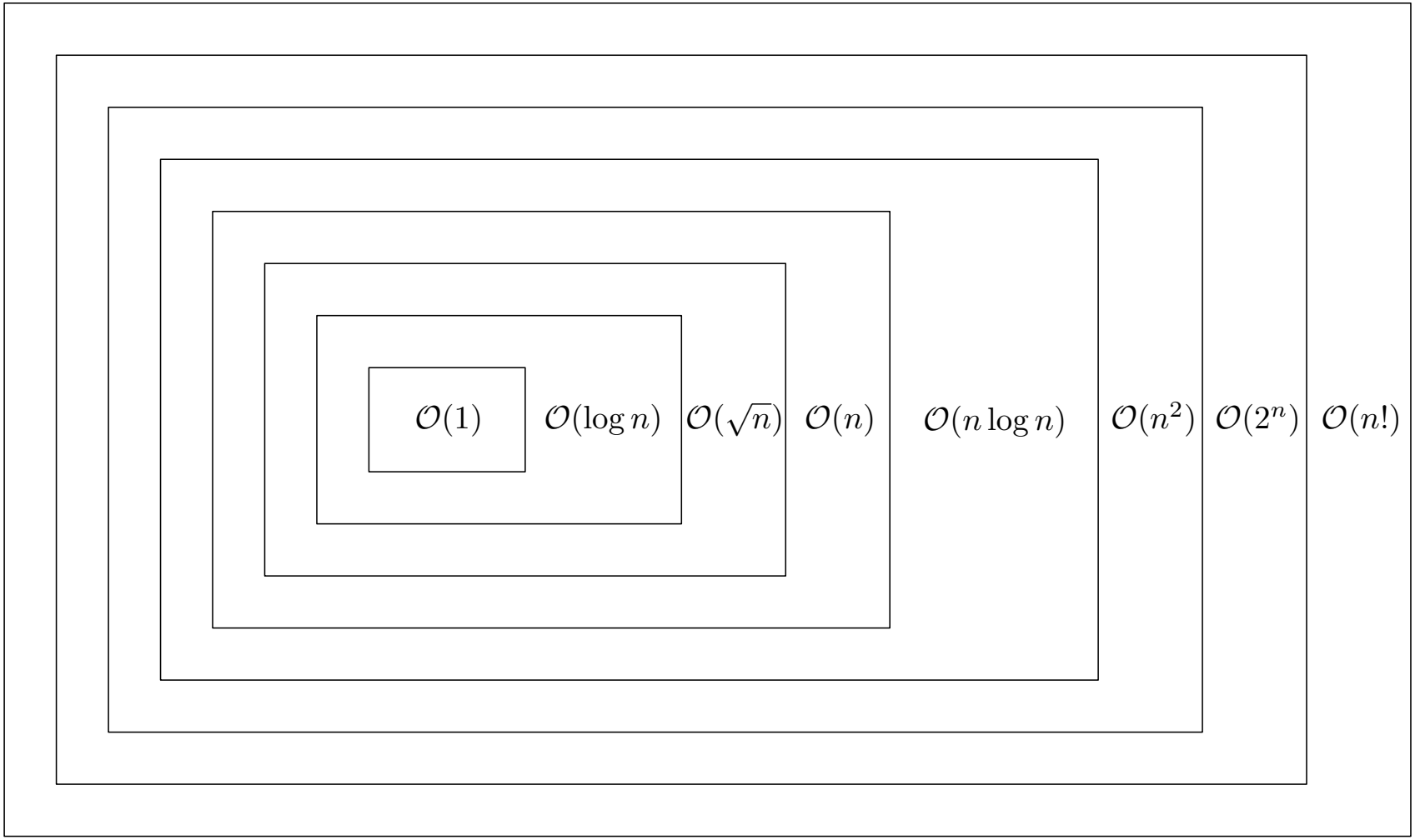
Asymptotic notations

Notation	Meaning
$\mathcal{O}(g(n))$ at most $g(n)$	Set of all functions with the same or lower order of growth as $g(n)$ $3n^2 \in \mathcal{O}(n^2)$, $n^2/17+n \in \mathcal{O}(n^2)$, $n(n-1)/2 \in \mathcal{O}(n^2)$ $n \in \mathcal{O}(n^2)$, $4\sqrt{n}+3\log^2 n \in \mathcal{O}(n^2)$, $2000 \in \mathcal{O}(n^2)$ $n^3 \notin \mathcal{O}(n^2)$, $0.001n^{\pi-1} \notin \mathcal{O}(n^2)$, $n^4+n+1 \notin \mathcal{O}(n^2)$
$\Omega(g(n))$ at least $g(n)$	Set of all functions with the same or higher order of growth as $g(n)$ $3n^2 \in \Omega(n^2)$, $n^2/17+n \in \Omega(n^2)$, $n(n-1)/2 \in \Omega(n^2)$ $n^3 \in \Omega(n^2)$, $0.001n^{\pi-1} \in \Omega(n^2)$, $n^4+n+1 \in \Omega(n^2)$ $n \notin \Omega(n^2)$, $4\sqrt{n}+3\log^2 n \notin \Omega(n^2)$, $2000 \notin \Omega(n^2)$
$\Theta(g(n))$ same as $g(n)$	Set of all functions with the same order of growth as $g(n)$ $3n^2 \in \Theta(n^2)$, $n^2/17+n \in \Theta(n^2)$, $n(n-1)/2 \in \Theta(n^2)$

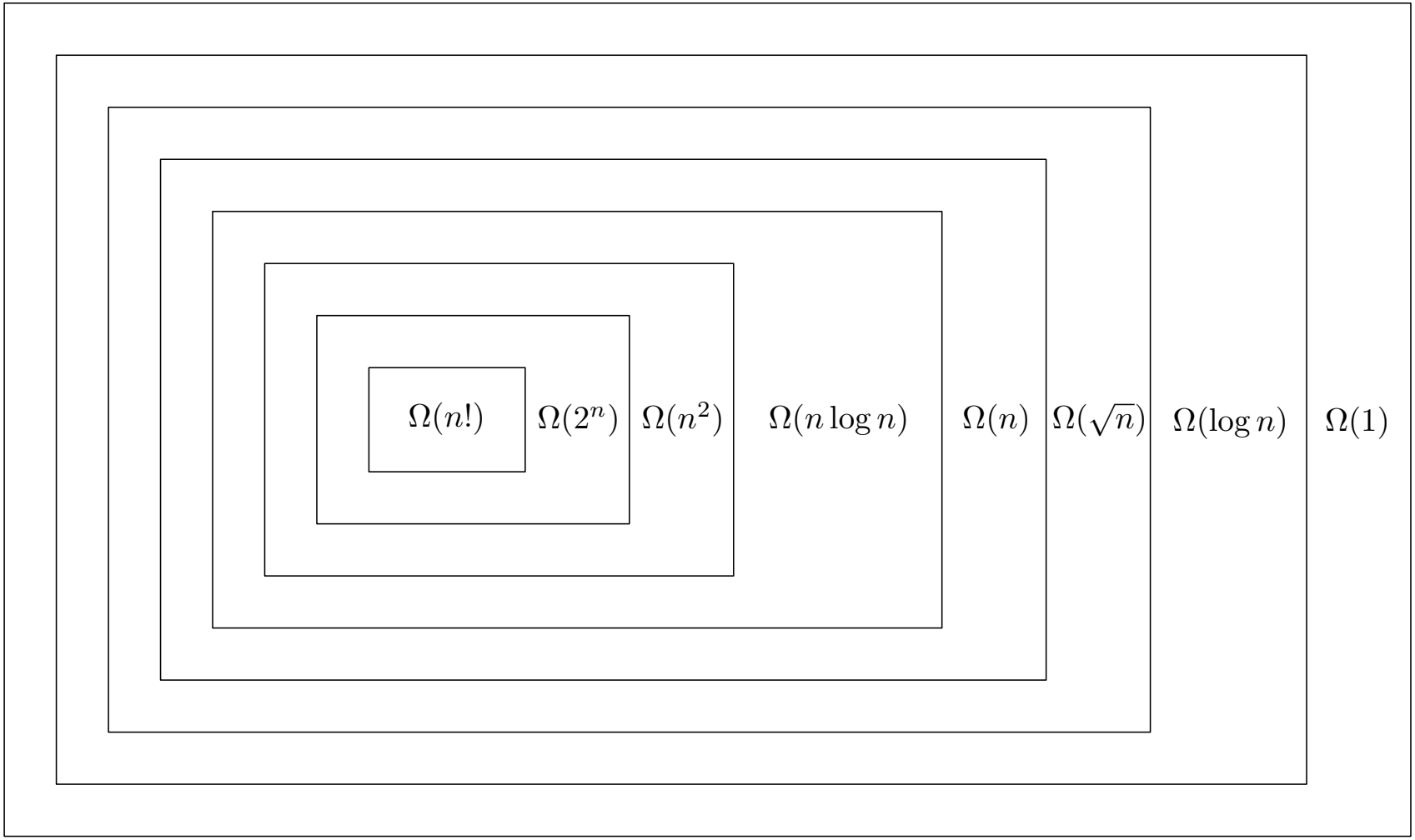
$\Theta\,()$ class

$\Theta(1)$	$\Theta(\log n)$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(2^n)$	$\Theta(n!)$

$\mathcal{O}\,()$ class



$\Omega\,()$ class


$\Omega(n!)$
$\Omega(2^n)$
$\Omega(n^2)$
$\Omega(n \log n)$
$\Omega(n)$
$\Omega(\sqrt{n})$
$\Omega(\log n)$
$\Omega(1)$

Asymptotic notations

Definition

Let us define the binary relations $\asymp, \prec, \preceq, \succ, \succeq$ over functions. For two functions $f(n)$ and $g(n)$:

- $f(n) \asymp g(n) \quad \Leftrightarrow \quad f(n) \in \Theta\left(g(n)\right)$
- $f(n) \preceq g(n) \quad \Leftrightarrow \quad f(n) \in \mathcal{O}\left(g(n)\right)$
- $f(n) \prec g(n) \quad \Leftrightarrow \quad f(n) \in o\left(g(n)\right)$
- $f(n) \succeq g(n) \quad \Leftrightarrow \quad f(n) \in \Omega\left(g(n)\right)$
- $f(n) \succ g(n) \quad \Leftrightarrow \quad f(n) \in \omega\left(g(n)\right)$

Asymptotic notations

Notation	Reflexivity	Symmetry	Transitivity
$\asymp$	✓	✓	✓
$\preceq$	✓	✗	✓
$\succeq$	✓	✗	✓
$\prec$	✗	✗	✓
$\succ$	✗	✗	✓

- $f(n) \preceq g(n) \Leftrightarrow g(n) \succeq f(n)$
- $f(n) \prec g(n) \Leftrightarrow g(n) \succ f(n)$
- $f(n) \asymp g(n) \Leftrightarrow f(n) \preceq g(n)$ and $f(n) \succeq g(n)$
- If $f_1(n) \preceq g_1(n)$ and $f_2(n) \preceq g_2(n)$, then $f_1(n) + f_2(n) \preceq \max(g_1(n), g_2(n))$
- How do you formally prove the propositions above?

Practice problems

Problem

Is it true that between any two functions, at least one of the five relations $\asymp, \preceq, \prec, \succeq, \succ$ holds between them?

Solution

- No.
- Counterexample:

  $f(n) = n$

  $$g(n) = \left\{ \begin{array}{ll} 1 & \text{if } n \text{ is odd,} \\ n^2 & \text{if } n \text{ is even.} \end{array} \right\}$$

Practice problems

Problem

Prove that for any two functions $f(n)$ and $g(n)$, $f(n) \asymp g(n)$ iff $f(n) \preceq g(n)$ and $f(n) \succeq g(n)$.

Solution

Part 1. $f(n) \asymp g(n) \Rightarrow f(n) \preceq g(n)$ and $f(n) \succeq g(n)$

- $0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n), \forall n \geq n_0$
- This implies

  $0 \leq c_1 \cdot g(n) \leq f(n), \forall n \geq n_0$ $\quad (f(n) \succeq g(n))$

  $0 \leq f(n) \leq c_2 \cdot g(n), \forall n \geq n_0$ $\quad (f(n) \preceq g(n))$

Part 2. $f(n) \preceq g(n)$ and $f(n) \succeq g(n) \Rightarrow f(n) \asymp g(n)$

- $0 \leq c' \cdot g(n) \leq f(n), \forall n \geq n'_0$ $\quad (f(n) \succeq g(n))$

  $0 \leq f(n) \leq c'' \cdot g(n), \forall n \geq n''_0$ $\quad (f(n) \preceq g(n))$
- This implies

  $0 \leq c' \cdot g(n) \leq f(n) \leq c'' \cdot g(n), \forall n \geq \max(n'_0, n''_0)$

Practice problems

Problem

Prove that for any two functions $f(n)$ and $g(n)$, $\max(f(n), g(n)) \asymp f(n) + g(n)$.

Solution

- Without loss of generality, we assume $\max(f(n), g(n)) = f(n)$.
- $\lim_{n\to\infty} \frac{f(n)}{f(n)} \leq \lim_{n\to\infty} \frac{f(n)+g(n)}{\max(f(n),g(n))} \leq \lim_{n\to\infty} \frac{2f(n)}{f(n)}$

$$\implies \lim_{n\to\infty} 1 \leq \lim_{n\to\infty} \frac{f(n)+g(n)}{\max(f(n),g(n))} \leq \lim_{n\to\infty} 2$$

$$\implies 1 \leq \lim_{n\to\infty} \frac{f(n)+g(n)}{\max(f(n),g(n))} \leq 2$$

$$\implies f(n) + g(n) \asymp \max(f(n), g(n))$$

Practice problems

Problem

Is it true that for any two functions $f(n)$ and $g(n)$ such that $f(n) - g(n)$ is asymptotically positive, it is the case that $\max(f(n), g(n)) \asymp f(n) - g(n)$.

Solution

- No.
- Counterexample.
  $f(n) = n^2 + n$
  $g(n) = n^2 + 1$
- Note: You cannot choose the same function $f(n) = g(n)$ as the difference is not asymptotically positive.

Practice problems

Problem

Prove that for any two functions $f(n)$ and $g(n)$ and any constant $k \in \mathbb{R}^+$, $f(n) \asymp g(n) \Leftrightarrow (f(n))^k \asymp (g(n))^k$

Solution

Part 1. $f(n) \asymp g(n) \Rightarrow (f(n))^k \asymp (g(n))^k$

- $0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
  $0 \leq c_1^k \cdot (g(n))^k \leq (f(n))^k \leq c_2^k \cdot (g(n))^k$

Part 2. $(f(n))^k \asymp (g(n))^k \Rightarrow f(n) \asymp g(n)$

- $0 \leq c_1 \cdot (g(n))^k \leq (f(n))^k \leq c_2 \cdot (g(n))^k$
  $0 \leq (c_1)^{\frac{1}{k}} \cdot g(n) \leq f(n) \leq (c_2)^{\frac{1}{k}} \cdot g(n)$

Example: Polynomial class

Problem

Show that if $f(n)$ is a polynomial of degree k, that is,
$T(n) = a_k n^k + a_{k-1} n^{k-1} \cdots + a_1 n + a_0$
and $a_k > 0$, then $T(n) \asymp n^k$.

Solution

For $n \geq 1$, we have $1 \leq n \leq n^2 \leq \cdots \leq n^k$.
We prove the theorem in two parts

1. Show that $T(n) \preceq n^k$
2. Show that $T(n) \succeq n^k$

Example: Polynomial class

Solution

Part 1. $T(n) \preceq n^k$

- $(a_k n^k + \cdots + a_1 n + a_0) \leq (|a_k| + \cdots + |a_1| + |a_0|)n^k = c_2 n^k$
- We have $T(n) \leq c_2 n^k$, where $c_2 > 0$, and $n_0 = 1$

Part 2. $T(n) \succeq n^k$

- Let $M = \max\left(\frac{|a_0|}{a_k}, \cdots, \frac{|a_{k-1}|}{a_k}\right)$, $c_1 = \frac{a_k}{2}$, and $n_0 = 2kM$
- $(a_k n^k + \cdots + a_1 n + a_0)$
  $= a_k n^k \left(1 + \frac{a_{k-1}}{a_k} \cdot \frac{1}{n} + \cdots + \frac{a_1}{a_k} \cdot \frac{1}{n^{k-1}} + \frac{a_0}{a_k} \cdot \frac{1}{n^k}\right)$
  $\geq a_k n^k \left(1 - \frac{M}{n_0} - \cdots - \frac{M}{n_0^{k-1}} - \frac{M}{n_0^k}\right)$
  $\geq a_k n^k \left(1 - \frac{kM}{n_0}\right)$
  $= c_1 n^k$
- We have $T(n) \geq c_1 n^k$, where $c_1 > 0$ and $n_0 > 0$

Example: Polynomial class

Problem

Show that if $f(n)$ is a polynomial of degree k, that is,
$T(n) = a_k n^k + a_{k-1} n^{k-1} \cdots + a_1 n + a_0$
and $a_k > 0$, then $T(n) \asymp n^k$.

Solution

$$\begin{aligned}
\lim_{n\to\infty} \frac{T(n)}{n^k} &= \lim_{n\to\infty} \left(\frac{a_k n^k + a_{k-1} n^{k-1} \cdots + a_1 n + a_0}{n^k} \right) \\
&= \lim_{n\to\infty} \left(a_k + \frac{a_{k-1}}{n} + \cdots + \frac{a_1}{n^{k-1}} + \frac{a_0}{n^k} \right) \\
&= a_k \qquad \text{(where, } 0 < a_k < \infty\text{)}
\end{aligned}$$

This implies $T(n) \asymp n^k$

Example: Quadratic function

Problem

Show that $\frac{1}{2}n(n-1) \asymp n^2$.

Solution

- Part 1. Show that $\frac{1}{2}n(n-1) \preceq n^2$
  $\frac{1}{2}n(n-1) = \frac{1}{2}n^2 - \frac{1}{2}n \leq \frac{1}{2}n^2$ for $n_0 = 1$
- Part 2. Show that $\frac{1}{2}n(n-1) \succeq n^2$
  $\frac{1}{2}n(n-1) = \frac{1}{2}n^2 - \frac{1}{2}n \geq \frac{1}{2}n^2 - \frac{1}{2}n\frac{1}{2}n = \frac{1}{4}n^2$ for $n_0 = 2$
- As $c_1 = \frac{1}{4}$, $c_2 = \frac{1}{2}$, and $n_0 = 2$, we have the result.

Solution

- $\lim_{n\to\infty} \frac{n(n-1)/2}{n^2} = \lim_{n\to\infty} \left(\frac{1}{2} - \frac{1}{2} \cdot \frac{1}{n}\right) = \frac{1}{2}$
- As limit is in $(0, \infty)$, we have the result

Sigma notation

$$\Sigma_{i \in \{i_1, i_2, \ldots, i_n\}} f(i) = f(i_1) + f(i_2) + \cdots + f(i_n)$$

Determining complexities from pseudocodes

Sequence-of-Statements

```
statement s1
statement s2
statement s3
```

$$\text{total time} = \text{time}(s_1) + \text{time}(s_2) + \text{time}(s_3)$$

If-Else-Ladder

```
if condition1 then
|   block b1
else if condition2 then
|   block b2
else
|   block b3
```

$$\text{total time} = \max(\ \text{time}(b_1), \text{time}(b_2), \text{time}(b_3)\ )$$

Determining complexities from pseudocodes

Loops

```
for i ← 1 to m do
    for j ← 1 to n do
        block b
```

$$\text{total time} = mn \times \text{time}(b)$$

(assuming block b takes the same time in every iteration)

Functions

```
for i ← 1 to m do
    for j ← 1 to n do
        F(i, j)                    // Suppose this takes Θ(ij) time
```

$$\text{total time} = \sum_{i=1}^{m} \sum_{j=1}^{n} \text{time}(\mathsf{F}(i,j))$$

Series sums

$$\Sigma_{i=1}^{n} 1 = 1 + 1 + 1 + \cdots + 1 = n$$

$$\Sigma_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

$$\Sigma_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\Sigma_{i=1}^{n} i^3 = 1^3 + 2^3 + 3^3 + \cdots + n^3 = (\Sigma_{i=1}^{n} i)^2 = \frac{n^2(n+1)^2}{4}$$

$$\Sigma_{i=0}^{n} 2^i = 2^0 + 2^1 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$$

$$\Sigma_{i=0}^{n} r^i = r^0 + r^1 + r^2 + \cdots + r^n = \frac{r^{n+1} - 1}{r - 1} \text{ for } r > 0$$

$$\Sigma_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln n$$

$$\Sigma_{i=0}^{n} \frac{1}{2^i} = \frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^3} + \cdots + \frac{1}{2^n} = 2 - \frac{1}{2^n}$$

$$\Sigma_{i=0}^{n} \frac{1}{r^i} = \frac{1}{r^0} + \frac{1}{r^1} + \frac{1}{r^3} + \cdots + \frac{1}{r^n} = \frac{1}{r^n} \cdot \frac{r^{n+1} - 1}{r - 1} \text{ for } r > 0$$

$$\Sigma_{i=2}^{n} \log i = \log 2 + \log 3 + \log 4 + \cdots + \log n = \Theta\left(n \log n\right)$$

$$\Sigma_{i=4}^{n} \log\log i = \log\log 4 + \log\log 5 + \log\log 6 + \cdots + \log\log n = \Theta\left(n \log\log n\right)$$

Important links: Brush up on your math skills

- Derivatives and differentiation rules
- List of integrals
- Maclaurin series
- List of mathematical series

Practice problems

Problem

Prove the following. Assume that the $\log$ function has base 2 unless explicitly mentioned otherwise.

$\frac{1}{n} \prec 1 \asymp 1 - \frac{1}{n} \asymp n^{\frac{1}{n}} \asymp n^{\frac{1}{\log n}} \asymp \sum_{i=1}^{n} \frac{1}{i^2} \asymp \sum_{i=1}^{n} \frac{1}{2^i} \prec \log\log n \asymp \sum_{\text{prime } p \leq n} \frac{1}{p} \prec \log n \asymp \log n^2 \asymp \sum_{i=1}^{n} \frac{1}{i} \prec \sqrt{n} \prec n \prec n \log n \asymp \sum_{i=1}^{n} \log i \asymp \log n! \prec n^2 \asymp \sum_{i=1}^{n} i \prec 2^n \asymp \sum_{i=0}^{n} {}^nC_i \prec n! \prec n^n \asymp (n+1)^n \prec n^{n+1}$

Practice problems

Problem

Prove that $\frac{1}{n} \prec 1$

Solution

- $\lim_{n\to\infty} \frac{1}{n} = 0$

Problem

Prove that $n^{\frac{1}{\log n}} \asymp 1$

Solution

- $\lim_{n\to\infty} n^{\frac{1}{\log n}}$
  $= \lim_{n\to\infty} (2^{\log n})^{\frac{1}{\log n}} = 2$

Practice problems

Problem

Prove that $n^{\frac{1}{n}} \asymp 1$

Solution

- $\lim_{n\to\infty} n^{\frac{1}{n}}$

$$
\begin{aligned}
&= \lim_{n\to\infty} e^{\ln(n^{\frac{1}{n}})} \\
&= \lim_{n\to\infty} e^{\frac{\ln n}{n}} && \text{(use Maclaurin series for } e^x\text{)} \\
&= \lim_{n\to\infty} \left(1 + \sum_{i=1}^{\infty} \frac{\left(\frac{\ln n}{n}\right)^i}{i!}\right) && \text{(remove } i!\text{ \& use upper bound)} \\
&\leq 1 + \lim_{n\to\infty} \sum_{i=1}^{\infty} \left(\frac{\ln n}{n}\right)^i \\
&= 1 + \lim_{x\to\infty} \sum_{i=1}^{\infty} \left(\frac{1}{x}\right)^i && \text{(set } x = \frac{n}{\ln n} > 1\text{)} \\
&= 1 + \lim_{x\to\infty} \frac{1}{x-1} && \text{(sum of geometric series)} \\
&= 1
\end{aligned}
$$

Practice problems

Problem

Prove that $\log n! \asymp n \log n$

Solution

- $\lim_{n\to\infty} \frac{\log n!}{n \log n} = \lim_{n\to\infty} \frac{\log(n\cdot(n-1)\cdot\dots\cdot 1)}{n \log n}$
  $< \lim_{n\to\infty} \frac{\sum_{i=1}^{\infty} \log i}{n \log n} = \lim_{n\to\infty} \frac{\frac{1}{\ln 2}\sum_{i=1}^{\infty} \ln i}{n \log n}$
  $\approx \lim_{n\to\infty} \frac{\frac{1}{\ln 2}\int_{i=1}^{\infty} \ln i}{n \log n} = \lim_{n\to\infty} \frac{\frac{1}{\ln 2}[x \ln x - x + c]_1^n}{n \log n}$
  $= \lim_{n\to\infty} \frac{\frac{1}{\ln 2}(n \ln n - n + 1)}{n \log n} \approx \lim_{n\to\infty} \frac{\frac{1}{\ln 2}(n \ln n - n)}{n \log n}$
  $= \lim_{n\to\infty} \frac{1}{\ln 2} \cdot \left(1 - \frac{1}{\log n}\right) = 1.44269504089\dots$

Practice problems

Problem

Prove that $1 \prec \log\log n$

Solution

- $\lim_{n\to\infty} \frac{1}{\log\log n} = 0$

Problem

Prove that $\log\log n \prec \log n$

Solution

- $\lim_{n\to\infty} \frac{\log\log n}{\log n}$
  $= \lim_{n\to\infty} \frac{\frac{1}{n \ln n \ln 2}}{\frac{1}{n \ln 2}}$ (use L'Hôpital's rule and Wolfram Alpha)
  $= \lim_{n\to\infty} \frac{1}{\ln n} = 0$

Practice problems

Problem

Evaluate the complexity of the LOOOOOOOOOOOP kernel. The many possible values for I_{start}, I_{end}, I_{incr}, J_{start}, J_{end}, and J_{incr} are given in the table. $start$ and end are abbreviated as s and e.

LOOOOOOOOOOOP(n)

for $i \leftarrow I_{start};\ i \leq I_{end};\ I_{incr}$ **do**
  for $j \leftarrow J_{start};\ j \leq J_{end};\ J_{incr}$ **do**
    do nothing

#	I_s	I_e	I_{incr}	J_s	J_e	J_{incr}
1	2	n	$i \leftarrow i+2$	2	n	$j \leftarrow j+2$
2	2	n	$i \leftarrow i+2$	2	n	$j \leftarrow j \times 2$
3	2	n	$i \leftarrow i+2$	2	n	$j \leftarrow j^2$
4	2	n	$i \leftarrow i \times 2$	2	n	$j \leftarrow j+2$
5	2	n	$i \leftarrow i \times 2$	2	n	$j \leftarrow j \times 2$
6	2	n	$i \leftarrow i \times 2$	2	n	$j \leftarrow j^2$
7	2	n	$i \leftarrow i^2$	2	n	$j \leftarrow j+2$
8	2	n	$i \leftarrow i^2$	2	n	$j \leftarrow j \times 2$
9	2	n	$i \leftarrow i^2$	2	n	$j \leftarrow j^2$
10	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow j+2$

#	I_s	I_e	I_{incr}	J_s	J_e	J_{incr}
11	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow j \times 2$
12	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow j^2$
13	2	n	$i \leftarrow i \times 2$	2	i	$j \leftarrow j+2$
14	2	n	$i \leftarrow i \times 2$	2	i	$j \leftarrow j \times 2$
15	2	n	$i \leftarrow i \times 2$	2	i	$j \leftarrow j^2$
16	2	n	$i \leftarrow i^2$	2	i	$j \leftarrow j+2$
17	2	n	$i \leftarrow i^2$	2	i	$j \leftarrow j \times 2$
18	2	n	$i \leftarrow i^2$	2	i	$j \leftarrow j^2$
19	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow i+j$
20	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow i \times j$

Practice problems

Problem

Kernel	I_{start}	I_{end}	I_{incr}	J_{start}	J_{end}	J_{incr}
1	2	n	$i \leftarrow i+2$	2	n	$j \leftarrow j+2$

Solution

Time

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \sum_{j \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} 1$$

$$= \left(\sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} 1\right) \cdot \left(\sum_{j \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} 1\right)$$

$$= \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{n}{2} \right\rfloor$$

$$= \left\lfloor \frac{n}{2} \right\rfloor^2$$

$$\in \Theta\left(n^2\right)$$

Practice problems

Problem

Kernel	I_{start}	I_{end}	I_{incr}	J_{start}	J_{end}	J_{incr}
2	2	n	$i \leftarrow i + 2$	2	n	$j \leftarrow j \times 2$

Solution

Time

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \sum_{j \in \{2^1,2^2,2^3,\ldots,2^{\lfloor \log n \rfloor}\}} 1$$

$$= \left(\sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} 1\right) \cdot \left(\sum_{j \in \{2^1,2^2,2^3,\ldots,2^{\lfloor \log n \rfloor}\}} 1\right)$$

$$= \left\lfloor \frac{n}{2} \right\rfloor \cdot \lfloor \log n \rfloor$$

$$\in \Theta\left(n \log n\right)$$

Practice problems

Problem

Kernel	I_{start}	I_{end}	I_{incr}	J_{start}	J_{end}	J_{incr}
3	2	n	$i \leftarrow i+2$	2	n	$j \leftarrow j^2$

Solution

Time

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \sum_{j \in \left\{2^{2^0}, 2^{2^1}, 2^{2^2}, \ldots, 2^{2^{\lfloor \log \lfloor \log n \rfloor \rfloor}}\right\}} 1$$

$$= \left(\sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} 1\right) \cdot \left(\sum_{j \in \left\{2^{2^0}, 2^{2^1}, 2^{2^2}, \ldots, 2^{2^{\lfloor \log \lfloor \log n \rfloor \rfloor}}\right\}} 1\right)$$

$$= \lfloor \frac{n}{2} \rfloor \cdot (\lfloor \log \lfloor \log n \rfloor \rfloor + 1)$$

$$\in \Theta\left(n \log \log n\right)$$

Practice problems

Problem

Kernel	I_{start}	I_{end}	I_{incr}	J_{start}	J_{end}	J_{incr}
10	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow j+2$

Solution

Time

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \sum_{j \in \{2,4,6,\ldots,2\lfloor \frac{i}{2} \rfloor\}} 1$$

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \left\lfloor \frac{i}{2} \right\rfloor$$

$$= 1 + 2 + 3 + \cdots + \left\lfloor \frac{n}{2} \right\rfloor$$

$$= \frac{\lfloor \frac{n}{2} \rfloor \cdot \left(\lfloor \frac{n}{2} \rfloor + 1 \right)}{2}$$

$$\in \Theta\left(n^2\right)$$

Practice problems

Problem

Kernel	I_{start}	I_{end}	I_{incr}	J_{start}	J_{end}	J_{incr}
11	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow j \times 2$

Solution

Time

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \sum_{j \in \{2^1,2^2,2^3,\ldots,2^{\lfloor \log i \rfloor}\}} 1$$

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \lfloor \log i \rfloor$$

$$= \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \lfloor \log(2i) \rfloor = \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \lfloor 1 + \log i \rfloor = \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} (1 + \lfloor \log i \rfloor)$$

$$= \lfloor \frac{n}{2} \rfloor + \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \lfloor \log i \rfloor$$

$$\leq \lfloor \frac{n}{2} \rfloor + \log\left(\lfloor \frac{n}{2} \rfloor !\right)$$

$$\in \Theta\left(n \log n\right)$$

Practice problems

Problem

Kernel	I_{start}	I_{end}	I_{incr}	J_{start}	J_{end}	J_{incr}
12	2	n	$i \leftarrow i+2$	2	i	$j \leftarrow j^2$

Solution

Time

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \sum_{j \in \left\{2^{2^0}, 2^{2^1}, 2^{2^2}, \ldots, 2^{2^{\lfloor \log \lfloor \log i \rfloor \rfloor}}\right\}} 1$$

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \left(\lfloor \log \lfloor \log i \rfloor \rfloor + 1\right)$$

$$= \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} \lfloor \log \lfloor \log i \rfloor \rfloor + \sum_{i \in \{2,4,6,\ldots,2\lfloor \frac{n}{2} \rfloor\}} 1$$

$$\leq \int_{x=2}^{n} \log \log x \cdot \mathrm{d}x + \left\lfloor \frac{n}{2} \right\rfloor$$

$$\leq \left(n \log \log n - \int \frac{1}{\log x} \mathrm{d}x\right) + \left\lfloor \frac{n}{2} \right\rfloor \qquad \left(\int u \cdot \mathrm{d}v = uv - \int v \cdot \mathrm{d}u\right)$$

$$\in \Theta\left(n \log \log n\right)$$