Contents

Section outline

1. **Asymptotic complexity of programs**
   - Asymptotic complexity
   - Big-O notation
   - Sample growth functions
   - Big-Omega notation
   - Determination of constants
   - Theta notation
   - Θ is an equivalence relation
   - Partial order relation induced by $\mathcal{O}$
   - Small-o notation: $o(g(n))$
   - Diagram of relation between Θ, $\mathcal{O}$ and o
   - Sample relations between functions
   - Small-ω notation: $\omega(g(n))$
   - Summary
   - Practice questions

Asymptotic complexity

- Suppose we determine that a program takes $8n + 5$ steps to solve a problem of size n
- What is the significance of the 8 and $+5$?
- As n gets large, the $+5$ becomes insignificant
- The 8 is inaccurate as different operations require varying amounts of time
- What is fundamental is that the time is *linear* in n
- *Asymptotic Complexity*: As n gets large, ignore all lower order terms and concentrate on the highest order term only

Asymptotic complexity (Contd.)

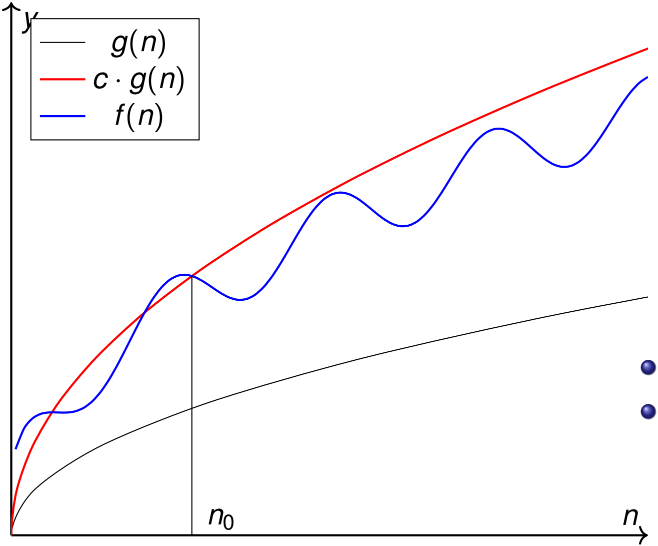
- $8n + 5$ is said to *grow asymptotically* like n
- So does $119n - 45$
- This gives us a simplified approximation of the complexity of the algorithm, leaving out details that become insignificant for larger input sizes

Big-O notation

- We have talked of $\mathcal{O}(n)$, $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ before
- The big-O notation is used to express the upper bound on a function, hence used to denote the worst case running time of a program
- If $f(n)$ and $g(n)$ are two functions then we can mathematically say:

  $f(n) \in \mathcal{O}(g(n))$ if there exist positive constants c and n_0 such that for all $n > n_0$, $0 \le f(n) \le cg(n)$
- $cg(n)$ dominates $f(n)$ for $n > n_0$ (for large n)
- This is read "$f(n)$ is order $g(n)$," or "$f(n)$ is big-O of $g(n)$"
- Loosely speaking, $g(n)$ grows at least as fast as $f(n)$
- Sometimes people also write $f(n) = \mathcal{O}(g(n))$, but that notation is misleading, as there is no straightforward equality involved
- This characterisation is not tight, if $f(n) \in \mathcal{O}(n)$, then $f(n) \in \mathcal{O}(n^2)$

Diagramatic representation of big-O



- $f(n) \in \mathcal{O}(g(n))$
- $\exists c, n_0 > 0, \forall n > n_0,$
  $0 \leq f(n) \leq c \cdot g(n)$

Sample growth functions

The functions below are given in ascending order:

$\mathcal{O}(k) = \mathcal{O}(1)$	Constant time
$\mathcal{O}(\log_b n) = \mathcal{O}(\log n)$	Logarithmic time
$\mathcal{O}(n)$	Linear time
$\mathcal{O}(n \log n)$	
$\mathcal{O}(n^2)$	Quadratic time
$\mathcal{O}(n^3)$	Cubic time
$\ldots$	
$\mathcal{O}(k^n)$	Exponential time
$\mathcal{O}(n!)$	Super exponential time

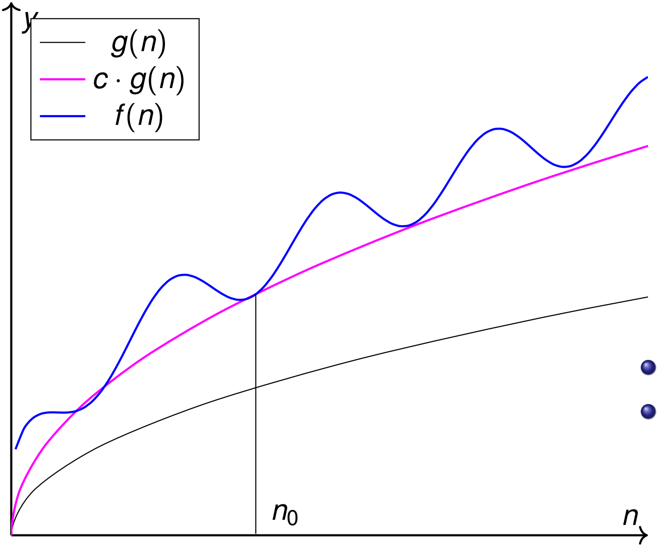
Big-Omega notation

- In matrix evaluation by Cramer's rule, the number of operations to be performed is worse that $n!$, if used with a naive determinant-finding algorithm
- The big-Omega notation is used to express the lower bound on a function
- If $f(n)$ and $g(n)$ are two functions then we can mathematically say:

  $f(n) \in \Omega(g(n))$ if there exist positive constants c and n_0 such that for all $n > n_0$ $0, \leq cg(n) \leq f(n)$

- $f(n)$ dominates $cg(n)$ for $n > n_0$ (for large n)
- Loosely speaking, $f(n)$ grows at least as fast as $g(n)$
- Sometimes people also write $f(n) = \Omega(g(n))$, but that notation is misleading, as there is no straightforward equality involved
- This characterisation is also not tight

Diagramatic representation of big-Omega



- $f(n) \in \Omega(g(n))$
- $\exists c, n_0 > 0, \forall n > n_0,$ $0 \leq c \cdot g(n) \leq f(n)$

Determination of constants

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}\left(n^3\right)$)

Determination of constants

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}(n^3)$)

- By definition, $T(n) \in \mathcal{O}(n^3)$ if $T(n) \leq c \cdot n^3$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \leq c \cdot n^3$ then $1 + \frac{20}{n^2} + \frac{1}{n^3} \leq c$
- Required condition holds for $n \geq n_0 = 1$ and $c \geq 22(= 1 + 20 + 1)$
- Larger values of n_0 result in smaller values c (for $n_0 = 10$, $c \geq 1.201$)

Determination of constants

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}(n^3)$)

- By definition, $T(n) \in \mathcal{O}(n^3)$ if $T(n) \leq c \cdot n^3$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \leq c \cdot n^3$ then $1 + \frac{20}{n^2} + \frac{1}{n^3} \leq c$
- Required condition holds for $n \geq n_0 = 1$ and $c \geq 22 (= 1 + 20 + 1)$
- Larger values of n_0 result in smaller values c (for $n_0 = 10$, $c \geq 1.201$)

Example ($T(n) = n^3 + 20n + 1 \notin \mathcal{O}(n^2)$)

Determination of constants

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}\left(n^3\right)$)

- By definition, $T(n) \in \mathcal{O}\left(n^3\right)$ if $T(n) \leq c \cdot n^3$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \leq c \cdot n^3$ then $1 + \frac{20}{n^2} + \frac{1}{n^3} \leq c$
- Required condition holds for $n \geq n_0 = 1$ and $c \geq 22 (= 1 + 20 + 1)$
- Larger values of n_0 result in smaller values c (for $n_0 = 10$, $c \geq 1.201$)

Example ($T(n) = n^3 + 20n + 1 \notin \mathcal{O}\left(n^2\right)$)

- By definition, $T(n) \in \mathcal{O}\left(n^2\right)$ if $T(n) \leq c \cdot n^2$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \leq c \cdot n^2$ then $n + \frac{20}{n} + \frac{1}{n^2} \leq c$ for $n \geq n_0$
- Clearly, required condition is insatisfiable, so $T(n) \notin \mathcal{O}\left(n^2\right)$

Determination of constants (contd.)

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}\left(n^4\right)$)

Determination of constants (contd.)

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}(n^4)$)

- By definition, $T(n) \in \mathcal{O}(n^4)$ if $T(n) \leq c \cdot n^4$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \leq c \cdot n^4$ then $\frac{1}{n} + \frac{20}{n^3} + \frac{1}{n^4} \leq c$
- Required condition holds for $n \geq n_0 = 1$ and $c \geq 22 (= 0.1 + 0.02 + 0.0001)$
- Larger values of n_0 result in smaller values for c (for $n_0 = 10$, $c \geq 0.1201$)

Determination of constants (contd.)

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}\left(n^4\right)$)

- By definition, $T(n) \in \mathcal{O}\left(n^4\right)$ if $T(n) \leq c \cdot n^4$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \leq c \cdot n^4$ then $\frac{1}{n} + \frac{20}{n^3} + \frac{1}{n^4} \leq c$
- Required condition holds for $n \geq n_0 = 1$ and $c \geq 22 (= 0.1 + 0.02 + 0.0001)$
- Larger values of n_0 result in smaller values for c (for $n_0 = 10$, $c \geq 0.1201$)

Example ($T(n) = n^3 + 20n + 1 \in \Omega(n^2)$)

Determination of constants (contd.)

Example ($T(n) = n^3 + 20n + 1 \in \mathcal{O}\left(n^4\right)$)

- By definition, $T(n) \in \mathcal{O}\left(n^4\right)$ if $T(n) \leq c \cdot n^4$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \leq c \cdot n^4$ then $\frac{1}{n} + \frac{20}{n^3} + \frac{1}{n^4} \leq c$
- Required condition holds for $n \geq n_0 = 1$ and $c \geq 22 (= 0.1 + 0.02 + 0.0001)$
- Larger values of n_0 result in smaller values for c (for $n_0 = 10$, $c \geq 0.1201$)

Example ($T(n) = n^3 + 20n + 1 \in \Omega(n^2)$)

- By definition, $T(n) \in \Omega(n^2)$ if $T(n) \geq c \cdot n^2$ for some $n \geq n_0$
- If $n^3 + 20n + 1 \geq c \cdot n^2$ then $n + \frac{20}{n} + \frac{1}{n^2} \geq c$ for $n \geq n_0$
- Required condition holds for $n \geq n_0 = \sqrt{20}$ and $c \leq 8.9 (\leq 2\sqrt{20} + 0.0025)$
- Larger values of n_0 result in larger values for c (for $n_0 = 20$, $c \leq 21$)

Theta notation

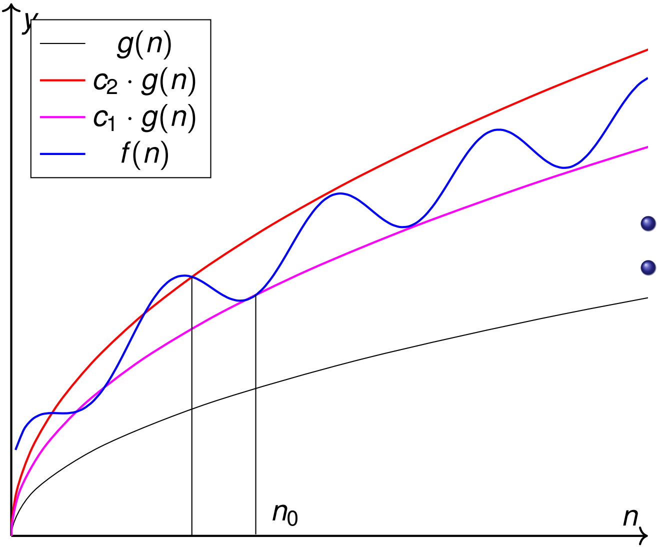
- The Theta notation is used to express the notion that a function $g(n)$ is a good (preferably simpler) characterisation of another function $f(n)$
- If $f(n)$ and $g(n)$ are two functions then we can mathematically say:

$f(n) \in \Theta(g(n))$ if there exist positive constants c_1, c_2 and n_0 such that for all $n > n_0$

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \tag{6.1}$$

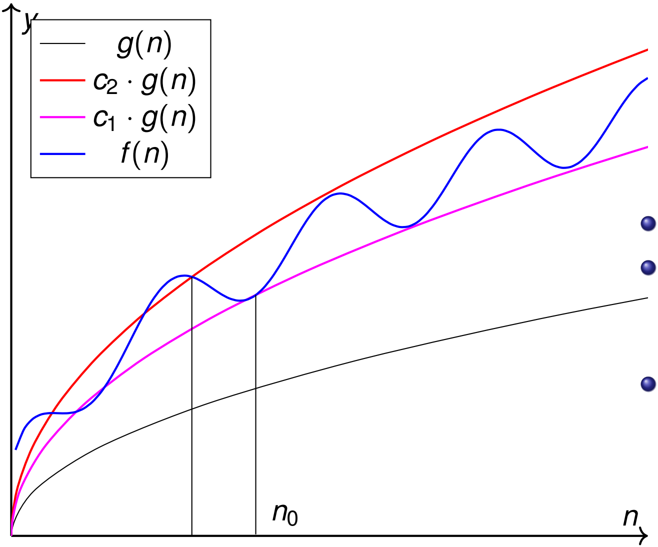
- Loosely speaking, $f(n)$ grows like $g(n)$
- Sometimes people also write $f(n) = \Theta(g(n))$, but that notation is misleading
- This characterisation is tight

Diagramatic representation of Theta



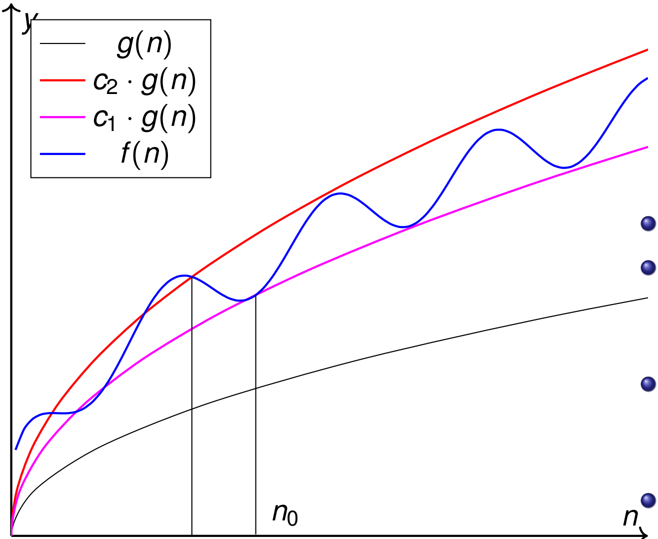
- $f(n) \in \Theta(g(n))$
- $\exists c_1, c_2, n_0 > 0, \forall n > n_0,\ 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

Diagramatic representation of Theta



- $f(n) \in \Theta(g(n))$
- $\exists c_1, c_2, n_0 > 0, \forall n > n_0,\ 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
- $f(n) \in \mathcal{O}(g(n)) \wedge f(n) \in \Omega(g(n)) \Rightarrow f(n) \in \Theta(g(n))$

Diagramatic representation of Theta



- $f(n) \in \Theta(g(n))$
- $\exists c_1, c_2, n_0 > 0, \forall n > n_0,\ 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
- $f(n) \in \mathcal{O}(g(n)) \wedge f(n) \in \Omega(g(n)) \Rightarrow f(n) \in \Theta(g(n))$
- $f(n) \in \Theta(g(n)) \Rightarrow g(n) \in \Theta(f(n))$

Θ is an equivalence relation

For a relation $\mathcal{R}$ on some set $\mathcal{F}$ be an equivalence relation, it needs to be:

- **reflexive**: so that for $f \in \mathcal{F}$, $f \mathcal{R} f$, ie: an item in $\mathcal{F}$ is related to itself by $\mathcal{R}$

Θ is an equivalence relation

For a relation $\mathcal{R}$ on some set $\mathcal{F}$ be an equivalence relation, it needs to be:

- **reflexive**: so that for $f \in \mathcal{F}$, $f\mathcal{R}f$, ie: an item in $\mathcal{F}$ is related to itself by $\mathcal{R}$
- **symmetric**: so that for $f_1, f_2 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \Rightarrow f_2\mathcal{R}f_1$

Θ is an equivalence relation

For a relation $\mathcal{R}$ on some set $\mathcal{F}$ be an equivalence relation, it needs to be:

- **reflexive**: so that for $f \in \mathcal{F}$, $f\mathcal{R}f$, ie: an item in $\mathcal{F}$ is related to itself by $\mathcal{R}$
- **symmetric**: so that for $f_1, f_2 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \Rightarrow f_2\mathcal{R}f_1$
- **transitive**: so that for $f_1, f_2, f_3 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \wedge f_2\mathcal{R}f_3 \Rightarrow f_1\mathcal{R}f_3$

Θ is an equivalence relation

For a relation $\mathcal{R}$ on some set $\mathcal{F}$ be an equivalence relation, it needs to be:

- **reflexive**: so that for $f \in \mathcal{F}$, $f\mathcal{R}f$, ie: an item in $\mathcal{F}$ is related to itself by $\mathcal{R}$
- **symmetric**: so that for $f_1, f_2 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \Rightarrow f_2\mathcal{R}f_1$
- **transitive**: so that for $f_1, f_2, f_3 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \wedge f_2\mathcal{R}f_3 \Rightarrow f_1\mathcal{R}f_3$

An equivalence relation partitions the underlying set into subsets so that

- elements in each subset are equivalent
- elements in different subsets are non-equivalent

Θ is an equivalence relation (contd.)

- We show that Θ is reflexive by substituting $f(x)$ for $g(x)$ in 6.1:

$$0 \leq c_1 f(n) \leq f(n) \leq c_2 f(n),\ \forall n \geq n_0$$

  which is satisfied for $c_1 = c_2 = 1$ and $n_0 = 0$
- Thus we conclude that Θ is **reflexive**

Θ is an equivalence relation (contd.)

- We show that Θ is reflexive by substituting $f(x)$ for $g(x)$ in 6.1:

$$0 \leq c_1 f(n) \leq f(n) \leq c_2 f(n), \ \forall n \geq n_0$$

  which is satisfied for $c_1 = c_2 = 1$ and $n_0 = 0$
- Thus we conclude that Θ is **reflexive**
- We need to show that Θ is symmetric

Θ is an equivalence relation (contd.)

- We show that Θ is reflexive by substituting $f(x)$ for $g(x)$ in 6.1:

$$0 \leq c_1 f(n) \leq f(n) \leq c_2 f(n), \ \forall n \geq n_0$$

  which is satisfied for $c_1 = c_2 = 1$ and $n_0 = 0$
- Thus we conclude that Θ is **reflexive**
- We need to show that Θ is symmetric
  - we divide the initial part of 6.1 by c_1 to get

$$0 \leq g(n) \leq \frac{1}{c_1} f(n), \ \forall n \geq n_0 \quad (7.1)$$

  - We divide the latter part of 6.1 by c_2 to get

$$0 \leq \frac{1}{c_2} f(n) \leq g(n), \ \forall n \geq n_0 \quad (7.2)$$

  - Combining 7.1 and 7.2, with $c_1' = \frac{1}{c_2}$ and $c_2' = \frac{1}{c_1}$, we get

$$0 \leq c_1' f(n) \leq g(n) \leq c_2' f(n), \ \forall n \geq n_0$$

- Thus we conclude that Θ is **symmetric**

Θ is an equivalence relation (contd.)

We show that Θ is transitive

- If $f(n) \in \Theta(g(n))$, then $\exists c_1 > 0, c_2 > 0, n_0 > 0$ such that,

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \ \forall n \geq n_0 \tag{7.3}$$

Θ is an equivalence relation (contd.)

We show that Θ is transitive

- If $f(n) \in \Theta(g(n))$, then $\exists c_1 > 0, c_2 > 0, n_0 > 0$ such that,

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \ \forall n \geq n_0 \quad (7.3)$$

- Likewise, if $g(n) \in \Theta(h(n))$, $\exists c'_1 > 0, c'_2 > 0, n'_0 > 0$ such that

$$0 \leq c'_1 h(n) \leq g(n) \leq c'_2 h(n), \ \forall n \geq n'_0 \quad (7.4)$$

Θ is an equivalence relation (contd.)

We show that Θ is transitive

- If $f(n) \in \Theta(g(n))$, then $\exists c_1 > 0, c_2 > 0, n_0 > 0$ such that,

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \ \forall n \geq n_0 \qquad (7.3)$$

- Likewise, if $g(n) \in \Theta(h(n))$, $\exists c'_1 > 0, c'_2 > 0, n'_0 > 0$ such that

$$0 \leq c'_1 h(n) \leq g(n) \leq c'_2 h(n), \ \forall n \geq n'_0 \qquad (7.4)$$

- Multiplying the first part of 7.4 by $c_1 > 1$, yields,

$$0 \leq c_1 c'_1 h(n) \leq c_1 g(n). \qquad (7.5)$$

Θ is an equivalence relation (contd.)

We show that Θ is transitive

- If $f(n) \in \Theta(g(n))$, then $\exists c_1 > 0, c_2 > 0, n_0 > 0$ such that,

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \ \forall n \geq n_0 \tag{7.3}$$

- Likewise, if $g(n) \in \Theta(h(n))$, $\exists c_1' > 0, c_2' > 0, n_0' > 0$ such that

$$0 \leq c_1' h(n) \leq g(n) \leq c_2' h(n), \ \forall n \geq n_0' \tag{7.4}$$

- Multiplying the first part of 7.4 by $c_1 > 1$, yields,

$$0 \leq c_1 c_1' h(n) \leq c_1 g(n). \tag{7.5}$$

- Multiplying the second part of 7.4 by $c_2 > 0$, yields,

$$c_2 g(n) \leq c_2 c_2' h(n) \tag{7.6}$$

Θ is an equivalence relation (contd.)

- Substituting 7.5 and 7.6 into 7.3 yields, $n \geq n_0,'' n_0'' = \max(n_0, n_0')$

$$0 \leq c_1 c_1' h(n) \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \leq c_2 c_2' h(n) \qquad (7.7)$$

- With $c_1'' = c_1 c_1', c_2'' = c_2 c_2'$,

$$0 \leq c_1'' h(n) \leq f(n) \leq c_2'' h(n), \ \forall n > n_0''$$

- Therefore, $f(n) \in \Theta(h(n))$, so Θ is **transitive**

Θ is an equivalence relation (contd.)

- Substituting 7.5 and 7.6 into 7.3 yields, $n \geq n_0,'' n_0'' = \max(n_0, n_0')$

$$0 \leq c_1 c_1' h(n) \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \leq c_2 c_2' h(n) \qquad (7.7)$$

- With $c_1'' = c_1 c_1', c_2'' = c_2 c_2'$,

$$0 \leq c_1'' h(n) \leq f(n) \leq c_2'' h(n), \ \forall n > n_0''$$

- Therefore, $f(n) \in \Theta(h(n))$, so Θ is **transitive**
- So, we can conclude that Θ is an **equivalence relation**

Partial order relation induced by $\mathcal{O}$

For a relation $\mathcal{R}$ on some set $\mathcal{F}$ be a partial order relation, it needs to be:

- **reflexive**: so that for $f \in \mathcal{F}$, $f\mathcal{R}f$, ie: an item in $\mathcal{F}$ is related to itself by $\mathcal{R}$

Partial order relation induced by $\mathcal{O}$

For a relation $\mathcal{R}$ on some set $\mathcal{F}$ be a partial order relation, it needs to be:

- **reflexive**: so that for $f \in \mathcal{F}$, $f\mathcal{R}f$, ie: an item in $\mathcal{F}$ is related to itself by $\mathcal{R}$
- **antisymmetric**: so that for $f_1, f_2 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \wedge f_2\mathcal{R}f_1 \Rightarrow f_1 = f_2$

Partial order relation induced by $\mathcal{O}$

For a relation $\mathcal{R}$ on some set $\mathcal{F}$ be a partial order relation, it needs to be:

- **reflexive**: so that for $f \in \mathcal{F}$, $f\mathcal{R}f$, ie: an item in $\mathcal{F}$ is related to itself by $\mathcal{R}$
- **antisymmetric**: so that for $f_1, f_2 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \wedge f_2\mathcal{R}f_1 \Rightarrow f_1 = f_2$
- **transitive**: so that for $f_1, f_2, f_3 \in \mathcal{F}$, $f_1\mathcal{R}f_2 \wedge f_2\mathcal{R}f_3 \Rightarrow f_1\mathcal{R}f_3$

The properties of $\mathcal{O}$ will now be examined

- $f(n) \in \mathcal{O}(f(n))$?

  Is it true that $\exists c > 0, n_0 > 0$ such that $\forall n \geq n_0$, $0 \leq f(n) \leq cf(n)$?

  Yes, for $c = 1$ and $n_0 = 1$, so $\mathcal{O}$ is reflexive

Partial order induced by $\mathcal{O}$ (contd.)

We show that $\mathcal{O}$ is transitive

- If $f(n) \in \mathcal{O}(g(n))$, then $\exists c > 0, n_0 > 0$ such that,

$$0 \leq f(n) \leq cg(n), \ \forall n \geq n_0 \tag{8.1}$$

Partial order induced by $\mathcal{O}$ (contd.)

We show that $\mathcal{O}$ is transitive

- If $f(n) \in \mathcal{O}(g(n))$, then $\exists c > 0, n_0 > 0$ such that,

$$0 \leq f(n) \leq cg(n), \ \forall n \geq n_0 \tag{8.1}$$

- Likewise, if $g(n) \in \mathcal{O}(h(n))$, $\exists c' > 0, n'_0 > 0$ such that

$$0 \leq g(n) \leq c'h(n), \ \forall n \geq n'_0 \tag{8.2}$$

Partial order induced by $\mathcal{O}$ (contd.)

We show that $\mathcal{O}$ is transitive

- If $f(n) \in \mathcal{O}(g(n))$, then $\exists c > 0, n_0 > 0$ such that,

$$0 \leq f(n) \leq cg(n), \ \forall n \geq n_0 \tag{8.1}$$

- Likewise, if $g(n) \in \mathcal{O}(h(n))$, $\exists c' > 0, n'_0 > 0$ such that

$$0 \leq g(n) \leq c'h(n), \ \forall n \geq n'_0 \tag{8.2}$$

- Multiplying 8.2 by c, yields,

$$0 \leq cg(n) \leq cc'h(n) \tag{8.3}$$

Partial order induced by $\mathcal{O}$ (contd.)

We show that $\mathcal{O}$ is transitive

- If $f(n) \in \mathcal{O}(g(n))$, then $\exists c > 0, n_0 > 0$ such that,

$$0 \leq f(n) \leq cg(n), \ \forall n \geq n_0 \tag{8.1}$$

- Likewise, if $g(n) \in \mathcal{O}(h(n))$, $\exists c' > 0, n'_0 > 0$ such that

$$0 \leq g(n) \leq c'h(n), \ \forall n \geq n'_0 \tag{8.2}$$

- Multiplying 8.2 by c, yields,

$$0 \leq cg(n) \leq cc'h(n) \tag{8.3}$$

- Combining 8.3 and 8.1, we have for $n > \max(n_0, n'_0)$:
  $0 \leq f(n) \leq cg(n) \leq cc'h(n)$, establishing that $\mathcal{O}$ is transitive

Partial order induced by $\mathcal{O}$ (contd.)

Is $\mathcal{O}$ antisymmetric?

- If $f(n) \in \mathcal{O}(g(n))$ and $g(n) \in \mathcal{O}(f(n))$ then $f(n) \in \Theta(g(n))$
  We have equivalence, but not equality

Partial order induced by $\mathcal{O}$ (contd.)

Is $\mathcal{O}$ antisymmetric?

- If $f(n) \in \mathcal{O}(g(n))$ and $g(n) \in \mathcal{O}(f(n))$ then $f(n) \in \Theta(g(n))$
  We have equivalence, but not equality
- Let $f \in \mathcal{O}(g)$
- Consider any $f' \in \Theta(f)$ and any $g' \in \Theta(g)$
- Clearly, $f' \in \mathcal{O}(f)$ and $g \in \mathcal{O}(g')$
- Alongwith $f \in \mathcal{O}(g)$, we have $f' \in \mathcal{O}(g')$ [by transitivity of $\mathcal{O}$]
- Thus, it makes sense to say $\Theta(f) \prec_{\mathcal{O}} \Theta(g)$ if $f \in \mathcal{O}(g)$

Partial order induced by $\mathcal{O}$ (contd.)

- It's now easy to see that $\prec_{\mathcal{O}}$ is a partial order induced by $\mathcal{O}$ satisfying

  reflexive $\Theta(f) \prec_{\mathcal{O}} \Theta(f)$

  antisymmetric if $\Theta(f) \prec_{\mathcal{O}} \Theta(g)$ and $\Theta(g) \prec_{\mathcal{O}} \Theta(f)$ then $\Theta(f) = \Theta(g)$

  transitive if $\Theta(f) \prec_{\mathcal{O}} \Theta(g)$ and $\Theta(g) \prec_{\mathcal{O}} \Theta(h)$ then $\Theta(f) = \Theta(h)$

- The partial order is induced on the Θ equivalence classes of f and g
- On similar lines Ω also induces are partial order (say $\prec_{\Omega}$)

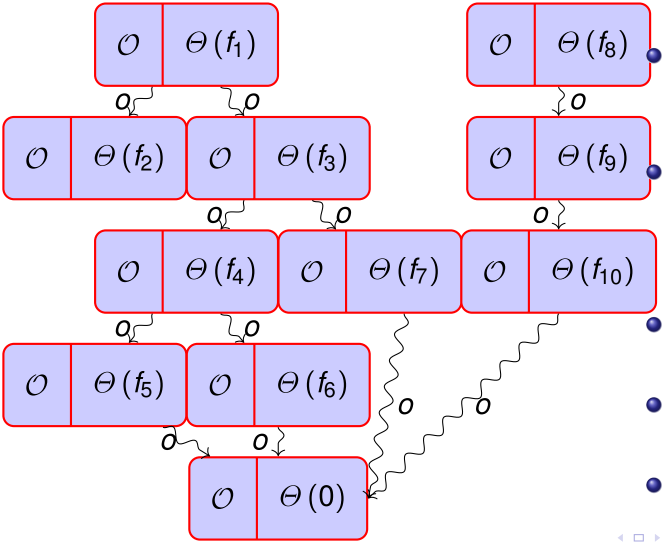
Small-o notation: $o(g(n))$

- A function $f(n)$ is said to be *asymptotically smaller than* $g(n)$ (denoted as $f(n) \in o(g(n))$), such that

$$o(g(n)) \triangleq \{f(n) : \forall \varepsilon > 0, \exists n_0 > 0, \forall n \geq n_0, 0 \leq f(n) < \varepsilon\, g(n)\}$$

- n_0 chosen in the above definition will usually depend on ε
- small-o differs from big-O by requiring the strict inequality to be satisfied for *any* value of ε (no matter how small)
- No way g can be scaled down to let f exceed εg asymtotically
- $$o(g(n)) = \left\{ f(n) : \lim_{n \to +\infty} \frac{f(n)}{g(n)} = 0 \right\}$$
- Eg: $n^2 \in o(n^3)$ (why?)

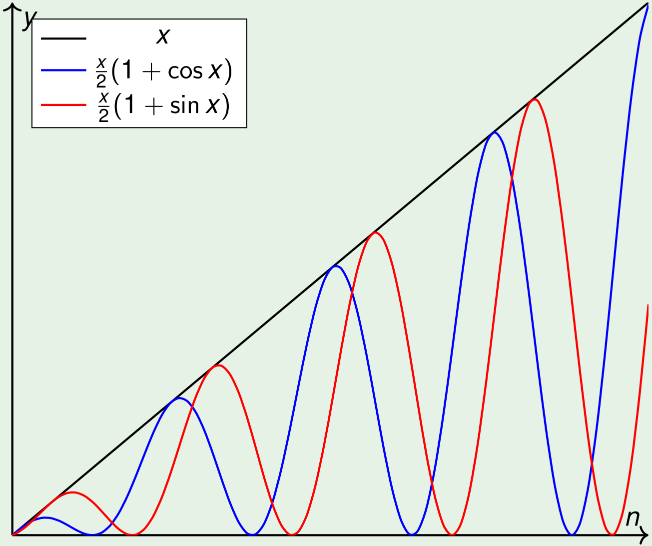
Diagram of relation between Θ, $\mathcal{O}$ and o



- Each box depicts the Θ class of f
- A f-node along with all its descendants constitutes $\mathcal{O}(f)$
- Descendants of a node are subset of the o class – to be seen next
- If $f_4(x) = x^2$ what could be f_5 and f_6?
- $0 \in o(f_5)$, $0 \in o(f_6)$, $0 \in o(f_7)$, etc.
- 0 is in small-o of many functions

Sample relations between functions

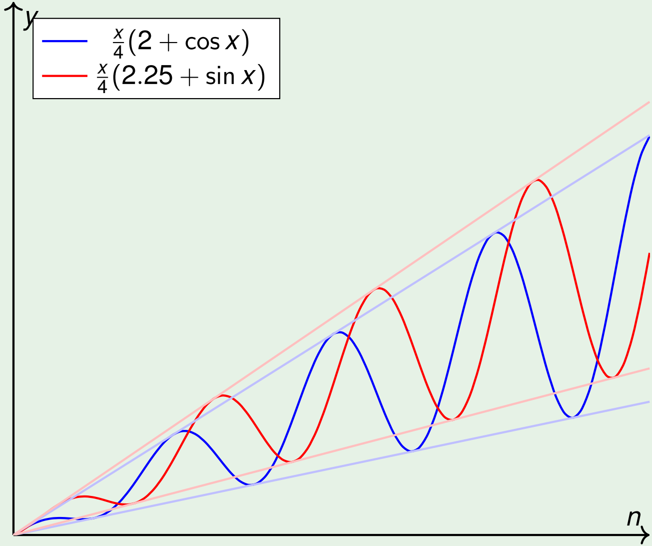
Example (Example of non-dominating functions)



- $f(n) \notin \mathcal{O}(g(n))$
- $g(n) \notin \mathcal{O}(f(n))$

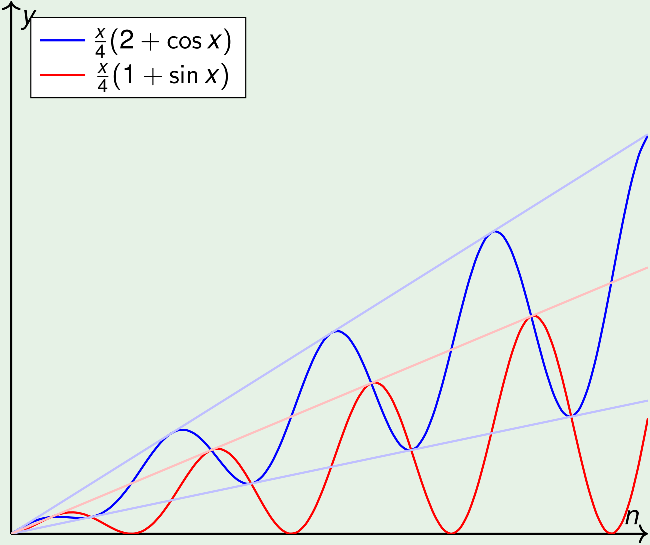
Another example

Example (What can be said about these functions)



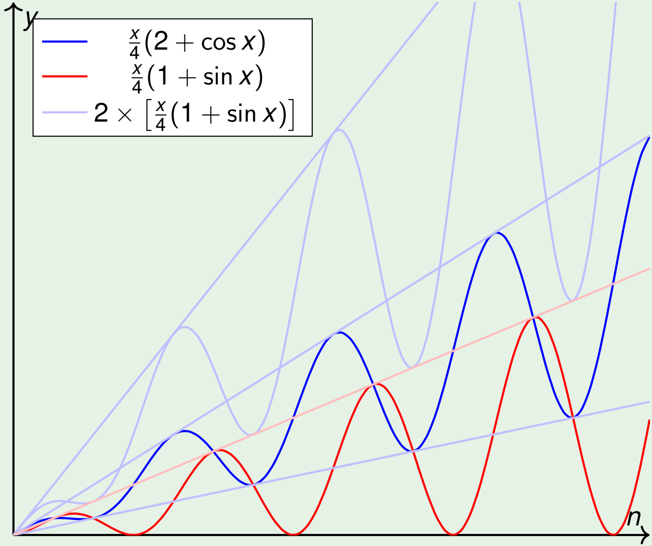
Yet another example

Example (What can be said about these functions)



Yet another example (contd.)

Example (What can be said about these functions)



Small-ω notation: $\omega(g(n))$

- A function $f(n)$ is said to be *asymptotically greater than* $g(n)$, if $f(n) \in \omega(g(n))$, where

$$\omega(g(n)) \triangleq \{f(n) : \forall c > 0, \exists n_0 > 0, \forall n \geq n_0, 0 \leq c\, g(n) < f(n)\}$$

- No matter how much g is scaled up, it never exceeds f, asymptotically
- $$\omega(g(n)) = \left\{f(n) : \lim_{n \to +\infty} \frac{f(n)}{g(h)} = +\infty\right\}$$

Summary

$\mathcal{O}(f)$ Functions that grow no faster than f

$\Omega(f)$ Functions that grow no slower than f

$\Theta(f)$ Functions that grow at the same rate as f

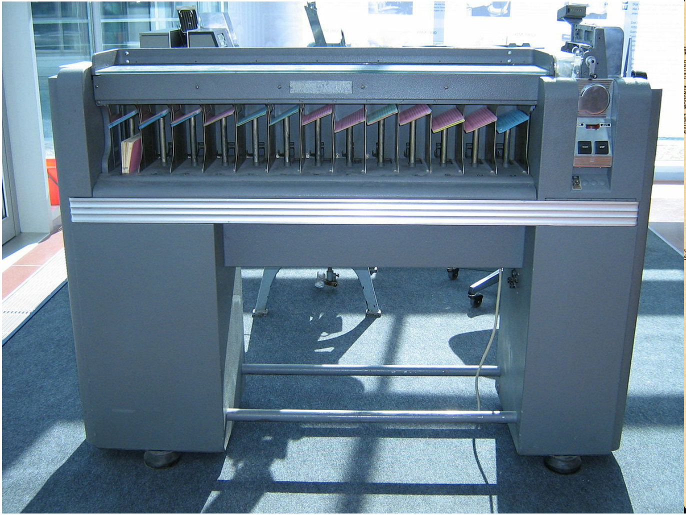
$o(f)$ Functions that grow slower than f

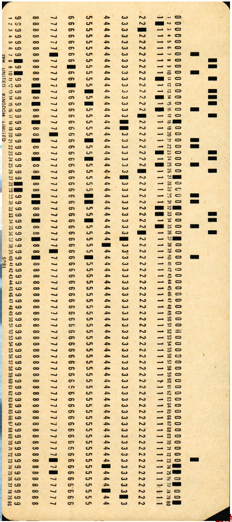
$$o(f) = \mathcal{O}(f) - \Theta(f)$$

$\omega(f)$ Functions that grow faster than f

$$\omega(f) = \Omega(f) - \Theta(f)$$

Practice questions





What's the time complexity of sorting with this card sorting machine?

Practice questions (contd.)

- Identify shortcomings of asymptotic analysis
- For the following code outline derive the worst-case asymptotic time complexity (in terms of n).

```
for (i=0; i<=n-1; i++) {
  for (j=i+1; j<=n-1; j++) {
    fixed length loop body
  }
}
```

- For each of the following pairs of functions $f_1(n)$ and $f_2(n)$ answer the following questions: **a** is $f_1(n) \in \mathcal{O}(f_2(n))$? **b** is $f_1(n) \in o(f_2(n))$? **c** is $f_1(n) \in \Theta(f_2(n))$? **d** is $f_1(n) \in \Omega f_2(n)$? **e** is $f_1(n) \in \omega(f_2(n))$?
  1. $f_1(n) = 6n^2$, $f_2(n) = n^2 \log n$
  2. $f_1(n) = \frac{3}{2}n^2 + 7n - 4$, $f_2(n) = 8n^2$
  3. $f_1(n) = n^4$, $f_2(n) = n^3 \log n^4$

Practice questions (contd.)

- If you were given two algorithms A_1 with time complexity $f_1(n)$ and A_2 with time complexity $f_2(n)$, which would you pick if your goal was to have the faster algorithm?
  You should justify your answer considering the definitions of $\mathcal{O}$, Θ, Ω and also the size of the inputs to be handled.
- Prove whether or not each of the following statements are true. Falsity should be established by giving a counterexample. Truth should be established wrt the formal definitions of $\mathcal{O}$, Ω and Θ. For all problems, assume $f(n) \geq 0$ and $g(n) \geq 0$.
  1. If $f(n) \in \mathcal{O}(g(n))$ then $g(n) \in \mathcal{O}(f(n))$
  2. $f(n) + g(n) \in \mathcal{O}(\max(f(n), g(n)))$
  3. If $f(n) \in \Omega g(n)$ then $g(n) \in \mathcal{O}(f(n))$

Practice questions (contd.)

Are each of the following true or false?

- $3n^2 + 10n\log n \in \mathcal{O}(n\log n)$
- $3n^2 + 10n\log n \in \Omega(n^2)$
- $3n^2 + 10n\log n \in \Theta(n^2)$
- $n\log n + n/2 \in \mathcal{O}(n)$
- $10\sqrt{n} + \log n \in \mathcal{O}(n)$
- $\sqrt{n} + \log n \in \mathcal{O}(\log n)$
- $\sqrt{n} + \log n \in \Theta(\log n)$
- $\sqrt{n} + \log n \in \Theta(n)$
- $2\sqrt{n} + \log n \in \Theta(\sqrt{n})$
- $\sqrt{n} + \log n \in \Omega(1)$
- $\sqrt{n} + \log n \in \Omega(\log n)$
- $\sqrt{n} + \log n \in \Omega(n)$