Unit #2: Complexity Theory and Asymptotic Analysis

CPSC 221: Algorithms and Data Structures

Lars Kotthoff[1]
larsko@cs.ubc.ca

[1]With material from Will Evans, Steve Wolfman, Alan Hu, Ed Knorr, and Kim Voll.

Asymptotic Behaviour

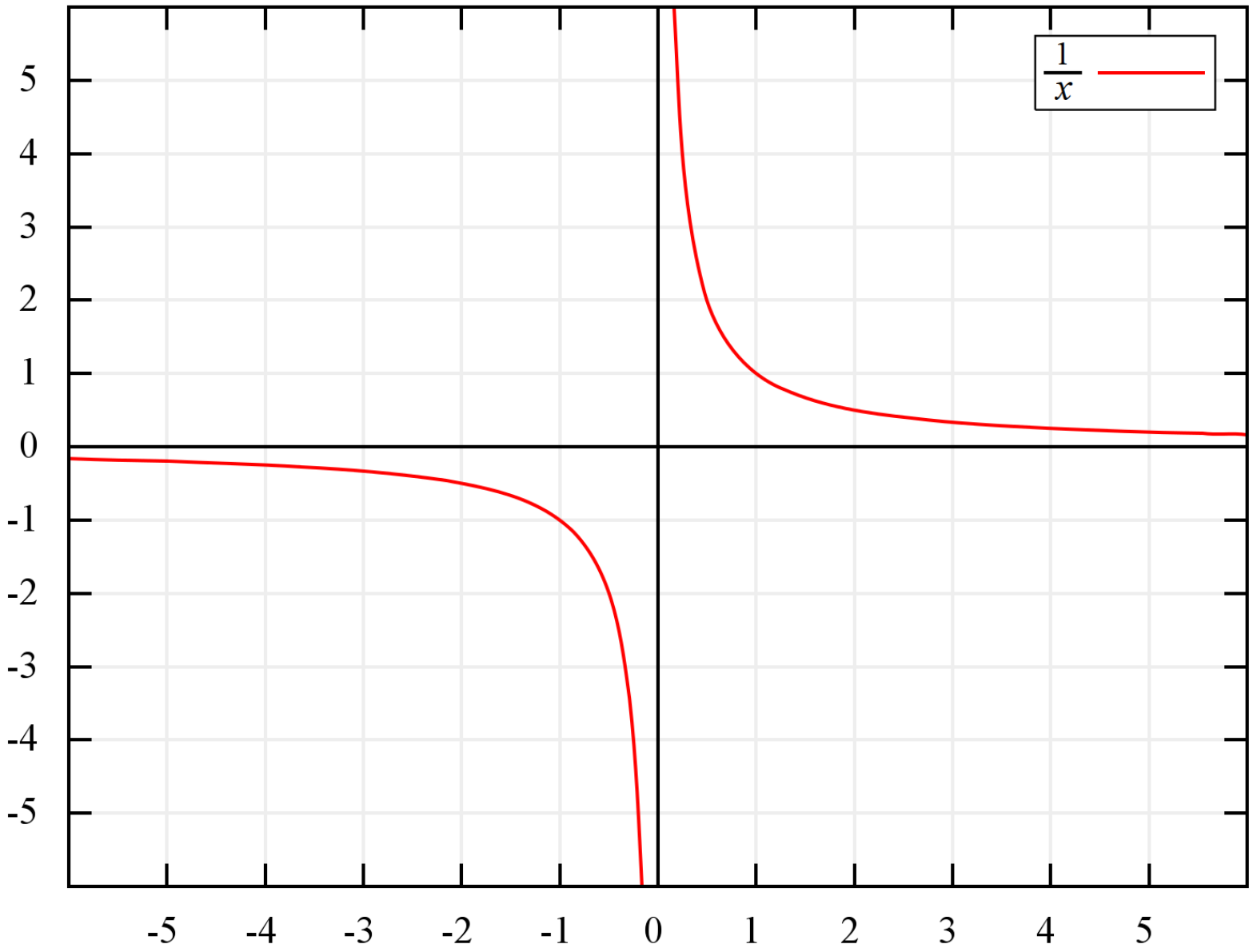


image from Wikipedia

Asymptotic Behaviour

- We measure runtime as a function of the input size n.
- We don't care about constants and constant factors.
- We are most interested what happens when n gets big.

Big-O Notation

$T(n) \in O(f(n))$ if there are positive constants c and n_0 such that

$$T(n) \leq cf(n) \text{ for all } n \geq n_0.$$

Asymptotic Notation

- $T(n) \in O(f(n))$ if there are positive constants c and n_0 such that $T(n) \leq cf(n)$ for all $n \geq n_0$.
- $T(n) \in \Omega(f(n))$ if there are positive constants c and n_0 such that $T(n) \geq cf(n)$ for all $n \geq n_0$.
- $T(n) \in \Theta(f(n))$ if $T(n) \in O(f(n))$ and $T(n) \in \Omega(f(n))$.
- $T(n) \in o(f(n))$ if for **any** positive constant c, there exists n_0 such that $T(n) < cf(n)$ for all $n \geq n_0$.
- $T(n) \in \omega(f(n))$ if for **any** positive constant c, there exists n_0 such that $T(n) > cf(n)$ for all $n \geq n_0$.

Examples

$$10,000n^2 + 25n \in \Theta(n^2)$$

$$10^{-10}n^2 \in \Theta(n^2)$$

$$n \log n \in O(n^2)$$

$$n \log n \in \Omega(n)$$

$$n^3 + 4 \in o(n^4)$$

$$n^3 + 4 \in \omega(n^2)$$

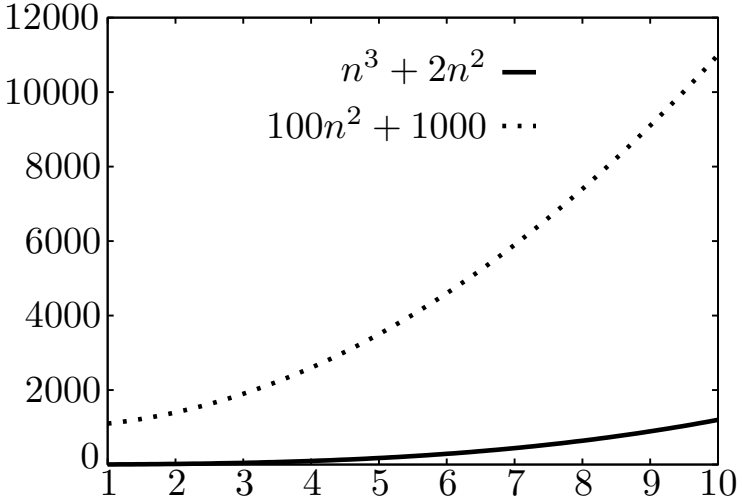
Analyzing Code

```
// Linear search
find(key, array)
  for i = 0 to length(array) - 1 do
    if array[i] == key
      return i
  return -1
```

4) How does $T(n) = 2n + 1$ behave asymptotically? What is the appropriate order notation? (O, o, Θ, Ω, ω?)

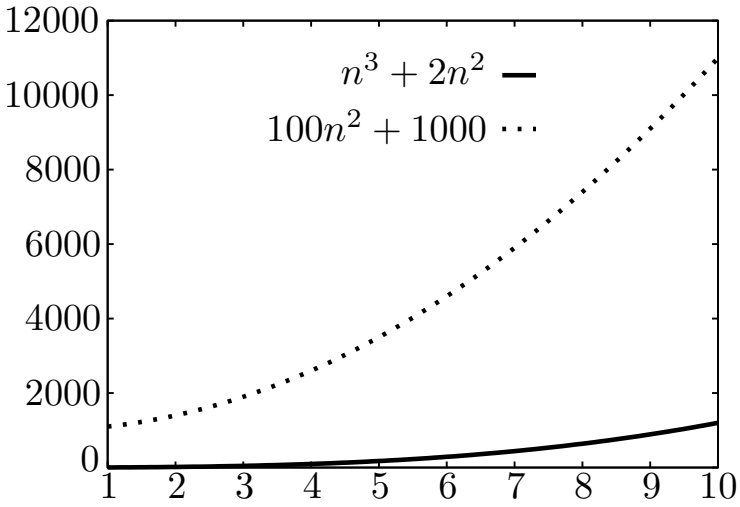
Asymptotically smaller?

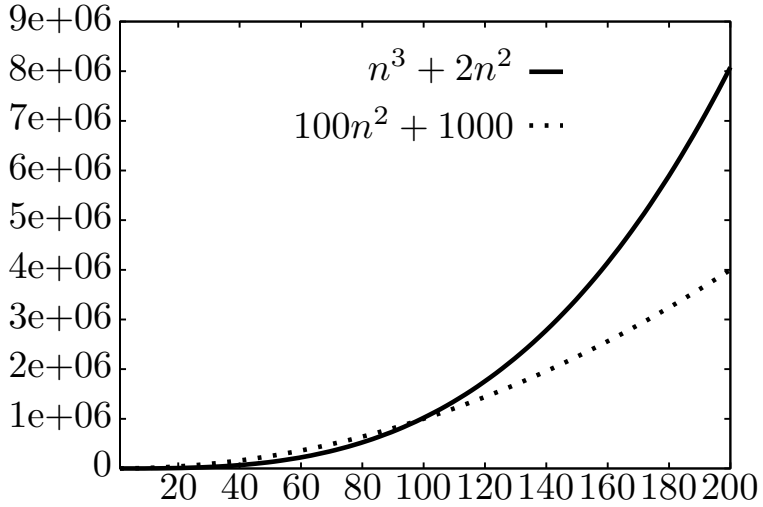
$$n^3 + 2n^2 \qquad \text{versus} \qquad 100n^2 + 1000$$



Asymptotically smaller?

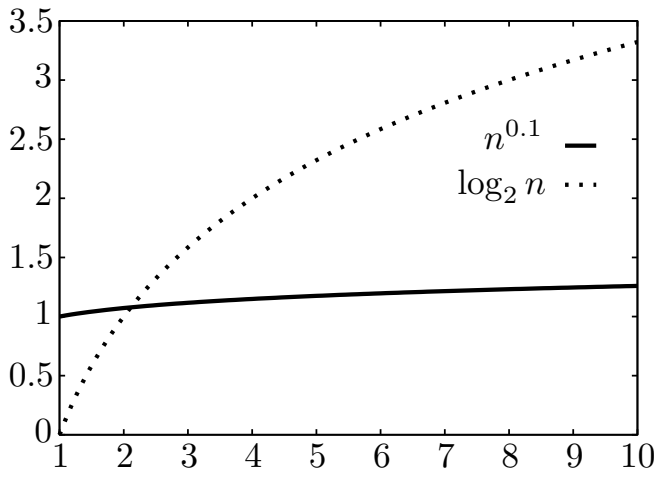
$$n^3 + 2n^2 \qquad \text{versus} \qquad 100n^2 + 1000$$





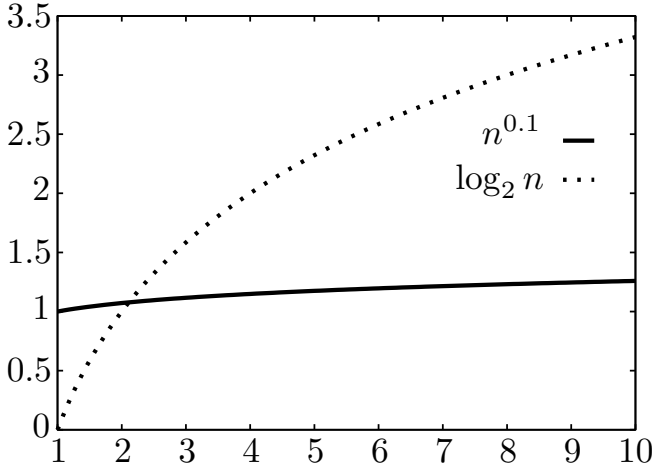
Asymptotically smaller?

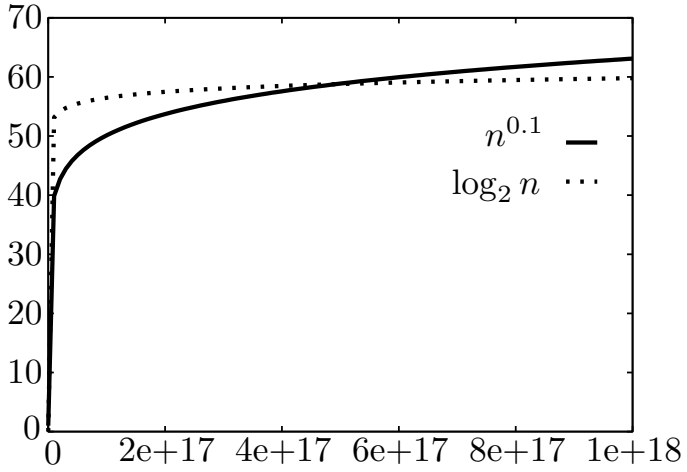
$$n^{0.1} \qquad \text{versus} \qquad \log_2 n$$



Asymptotically smaller?

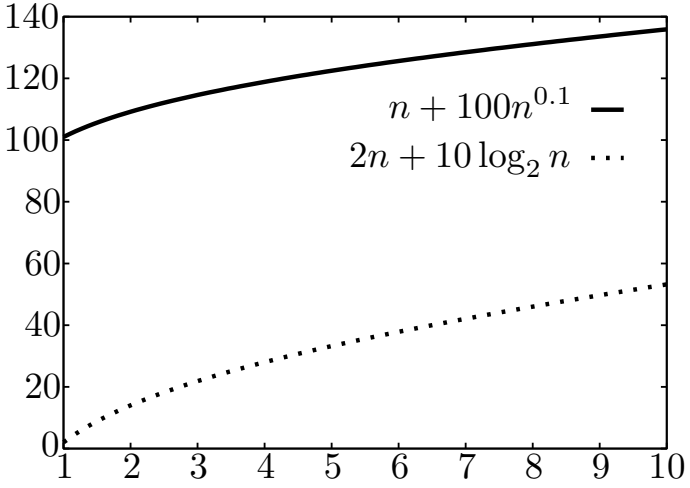
$$n^{0.1} \qquad \text{versus} \qquad \log_2 n$$





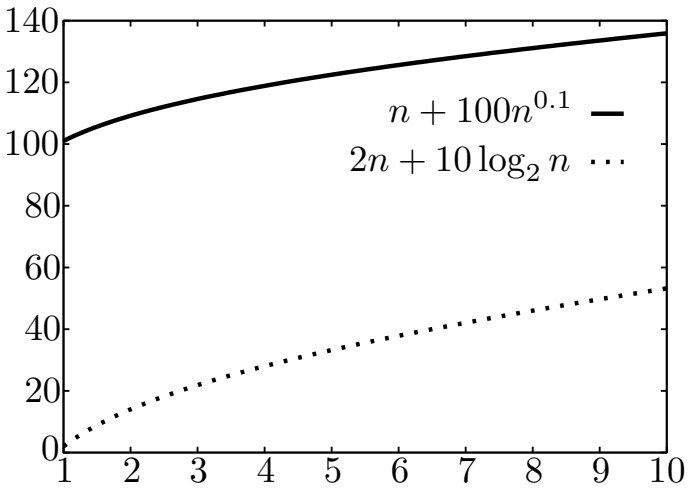
Asymptotically smaller?

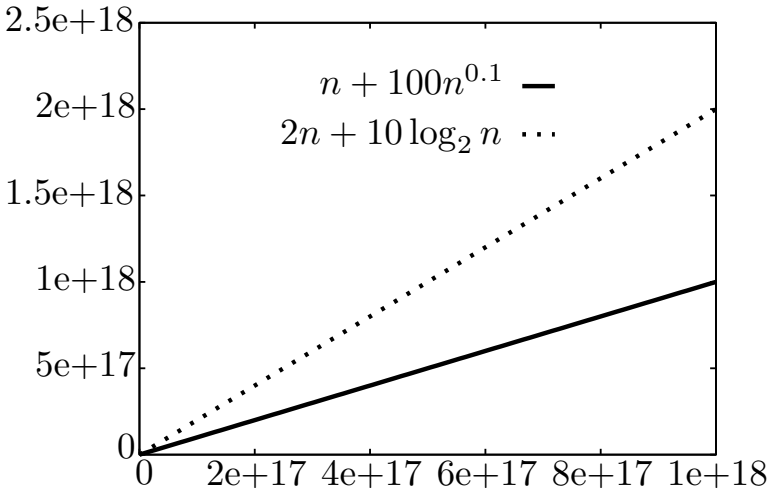
$$n + 100n^{0.1} \qquad \text{versus} \qquad 2n + 10\log_2 n$$



Asymptotically smaller?

$$n + 100n^{0.1} \qquad \text{versus} \qquad 2n + 10\log_2 n$$





Typical asymptotics

Tractable

- constant: $\Theta(1)$
- logarithmic: $\Theta(\log n)$ ($\log_b n$, $\log n^2 \in \Theta(\log n)$)
- poly-log: $\Theta(log^k n)$ ($\log^k n \equiv (\log n)^k$)
- linear: $\Theta(n)$
- log-linear: $\Theta(n \log n)$
- superlinear: $\Theta(n^{1+c})$ (c is a constant > 0)
- quadratic: $\Theta(n^2)$
- cubic: $\Theta(n^3)$
- polynomial: $\Theta(n^k)$ (k is a constant)

Intractable

- exponential: $\Theta(c^n)$ (c is a constant > 1)

Sample asymptotic relations

- $\{1, \log n, n^{0.9}, n, 100n\} \subset O(n)$
- $\{n, n \log n, n^2, 2^n\} \subset \Omega(n)$
- $\{n, 100n, n + \log n\} \subset \Theta(n)$
- $\{1, \log n, n^{0.9}\} \subset o(n)$
- $\{n \log n, n^2, 2^n\} \subset \omega(n)$

Analyzing Code

- single operations: constant time
- consecutive operations: sum operation times
- conditionals: condition time plus max of branch times
- loops: sum of loop-body times
- function call: time for function

Above all, use your head!

Runtime example #1

```
for i = 1 to n do
  for j = 1 to n do
    sum = sum + 1
```

Runtime example #2

```
i = 1
while i < n do
  for j = i to n do
    sum = sum + 1
  i++
```