CS101
Data Structures and Algorithms

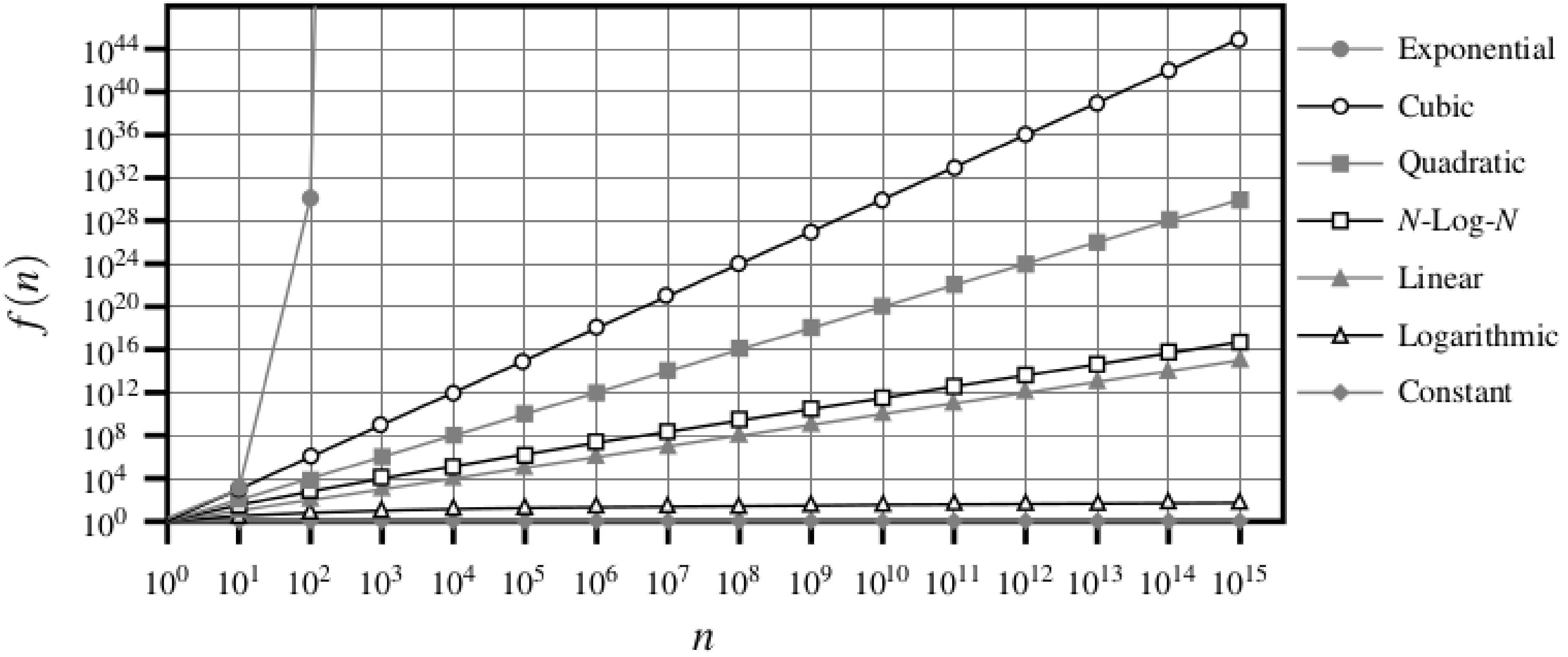
Lecture 04

Complexity Measures

Seven Typical Measures / Functions

- Constant – c
- Linear – n
- Quadratic – n^2
- Cubic – n^3
- Logarithmic – $\log n$
- Linear times Logarithmic – $n\log n$
- Exponential – b^n

Comparing the growth rates



Asymptotic Analysis

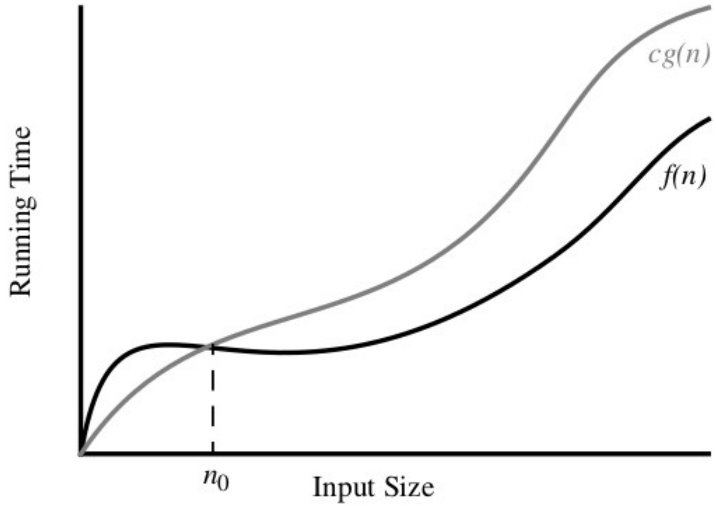
- Analyzing the algorithm performance for increasing input size is called Asymptotic Analysis (AA).
- AA is a static analysis, that is analysis based on the design and not run of the algorithm
- In AA we can afford to ignore the algorithm performance for initial values of input size.
-

Complexity Measures (Tractable)

- Big-Oh – O
- Small-Oh – o
- Big-Omega – Ω
- Small-Omega – ω
- Theta – θ

Notation Explained

- f is big-oh of g, that is f ~ O(g), or $f(n) \in O(n)$
- There exists $n_0 > 0$, $c > 0$ such that $f(n) <= cg(n)$ for all $n>=n_0$



Notation Explained

- f is big-Omega of g, that is $f \sim \Omega(g)$
  - There exists $n_0 > 0$, $c > 0$ such that $f(n) >= cg(n)$ for all $n>=n_0$
- f is small-oh of g, that is $f \sim o(g)$
  - There exists $n_0 > 0$, $c > 0$ such that $f(n) < cg(n)$ for all $n>=n_0$
- f is small-omega of g, that is $f \sim \omega(g)$
  - There exists $n_0 > 0$, $c > 0$ such that $f(n) > cg(n)$ for all $n>=n_0$
- f is Big-Theta or Theta of g, that is $f \sim \theta(g)$
  - There exists $n_0 > 0$, $a, b > 0$ such that $a.g(n) < f(n) < b.g(n)$ for all $n>=n_0$