reading: just browse number rep: understand main theorem

CSC165 fall 2017

begin algorithm analysis

Danny Heap
csc16517f@cs.toronto.edu
BA4270 (behind elevators)
Web page:
http://www.teach.cs.toronto.edu/~heap/165/F17/
416-978-5899

Using Course notes: more Induction
Outline

notes
def f(list_):
    for i in list_:
        print(i)

How much time does this take?

depends (linearly) on
depend on processor
depend on language
Size of each i

measure versus
"wall clock"

available

len(list_) = n

depends on print function
depends on print function
independent of n

Ram
assumptions, assumptions...

- “steps” independent of input size
  \( X \gg Y \quad X \times Y \quad \text{len(s)} \)
  \( a = 15 \quad \text{constant - often 1} \)

- ignore constant factors
  Comparison up to constant factor

- ignore “noise” for small input → near size 0

We care about growth rate of time consumption
formalizing assumptions

\[ f: \mathbb{N} \rightarrow \mathbb{R}^+ \]

- \( f \) absolutely dominates \( g \)
  
  Let \( f, g: \mathbb{N} \rightarrow \mathbb{R}^+ \), \( \forall n \in \mathbb{N} \), \( f(n) \geq g(n) \)

- \( f \) dominates \( g \) up to a constant factor
  
  Let \( f, g: \mathbb{N} \rightarrow \mathbb{R}^+ \), \( \exists c \in \mathbb{R}^+ \), \( \forall n \in \mathbb{N} \), \( cf(n) \geq g(n) \)

- \( f \) eventually dominates \( g \) up to a constant factor
  
  Let \( f, g: \mathbb{N} \rightarrow \mathbb{R}^+ \), \( \exists c \in \mathbb{R}^+ \), \( \exists n_0 \in \mathbb{R}^+ \), \( \forall n \in \mathbb{N} \), \( n \geq n_0 \Rightarrow cf(n) \geq g(n) \)

What should domain and range of \( f, g \) be?

\[ f(n) = 10n + 13,000 \]
\[ g(n) = 157n + 13,000 \]
big-Oh, big-Omega, big-Theta

... and you’re started on the Greek alphabet...

"g is big-oh", \( g \in \Theta(f) \): \( g \in \exists h : \mathbb{N} \rightarrow \mathbb{R}^+ \mid \exists c, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow h(n) \leq c f(n) \)

"g is big Omega of f"

\( g \in \Omega(f) \): \( g \in \exists h : \mathbb{N} \rightarrow \mathbb{R}^+ \mid \exists c, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow h(n) \geq c f(n) \)

"g is big Theta of f"

\( g \in \Theta(f) \): \( g \in \exists h : \mathbb{N} \rightarrow \mathbb{R}^+ \mid \exists c_1, c_2, n_0 \in \mathbb{R}, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow h(n) \geq c_1 f(n) \land h(n) \leq c_2 f(n) \)
big-Oh hierarchy

$\log_a n$ versus $\log_b n$ (logarithmic)

$n^a$ versus $n^b$ (polynomial)

$a^n$ versus $b^n$ (exponential)

$\log_a n$ versus $n^a$

$n^a$ versus $b^n$

explore!

\[
10^\log x = x = 2^{\log_2 x}
\]

\[
\log_{10} x = \log_{10^2} x \cdot \log_2 x
\]

\[
\frac{\log_{10} x}{\log_{10} 2} = \log_2 x
\]