Topic 1: Growth of Functions, Summations

(CLRS 3, Appendix A)

CPS 230, Fall 2001

1 Algorithms matter!

- Sort 10 million integers on
  - 1 GHZ computer (1000 million instructions per second) using $2n^2$ algorithm.
  - 100 MHz personal computer (100 million instructions per second) using $50n \log n$ algorithm.
- Computer : $\frac{2 \cdot (10^7)^2 \ inst.}{10^9 \ inst. \ per \ second} = 200000$ seconds ≈ 55 hours.
- Personal computer : $\frac{50 \cdot 10^7 \cdot \log 10^7 \ inst.}{10^8 \ inst. \ per \ second} < \frac{50 \cdot 10^7 \cdot 7 \cdot 3}{10^8} = 5 \cdot 7 \cdot 3 = 105$ seconds.

2 Asymptotic Growth

- In the insertion-sort example we discussed that when analyzing algorithms we are
  - interested in worst-case running time as function of input size n
  - not interested in exact constants in bound
  - not interested in lower order terms
- A good reason for not caring about constants and lower order terms is that the RAM model is not completely realistic anyway (not all operations cost the same)

$\Downarrow$

- We want to express *rate of growth* of standard functions:
  - the leading term with respect to n
  - ignoring constants in front of it

$$\boxed{\begin{array}{l} k_1 n + k_2 \asymp n \\ k_1 n \log n \asymp n \log n \\ k_1 n^2 + k_2 n + k_3 \asymp n^2 \end{array}}$$

- We also want to formalize e.g. that a $n \log n$ algorithms is better than a n^2 algorithm.

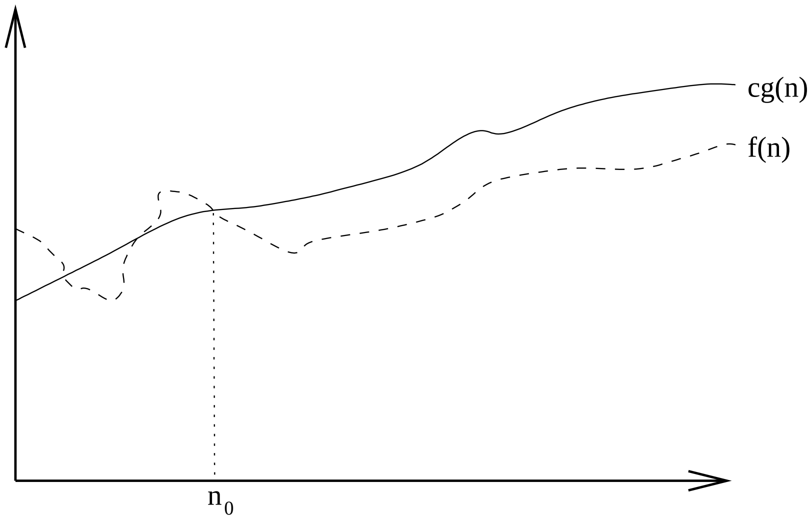
$\Downarrow$

- O-notation (Big-O)
  - you have probably all seen it intuitively defined but we will now define it more carefully.

2.1 O-notation (Big-O)

$O(g(n)) = \{f(n) : \exists\ c, n_0 > 0$ such that $|f(n)| \leq c|g(n)|\ \forall n \geq n_0\}$

- $O(\cdot)$ is used to asymptotically *upper bound* a function.
- $O(\cdot)$ is used to bound *worst-case* running time.

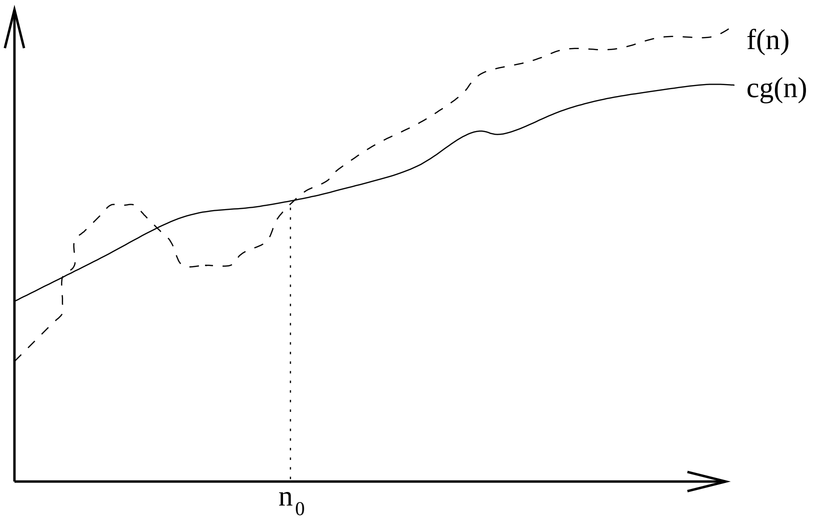


- Examples:
  - $1/3n^2 - 3n \in O(n^2)$ because $1/3n^2 - 3n \leq cn^2$ if $c \geq 1/3 - 3/n$ which holds for $c = 1/3$ and $n > 1$.
  - $k_1n^2 + k_2n + k_3 \in O(n^2)$ because $k_1n^2 + k_2n + k_3 < (k_1 + |k_2| + |k_3|)n^2$ and for $c > k_1 + |k_2| + |k_3|$ and $n \geq 1$, $k_1n^2 + k_2n + k_3 < cn^2$.
  - $k_1n^2 + k_2n + k_3 \in O(n^3)$ as $k_1n^2 + k_2n + k_3 < (k_1 + k_2 + k_3)n^3$ (Upper bound!).
- Note:
  - When we say "the running time is $O(n^2)$" we mean that the worst-case running time is $O(n^2)$ — best case might be better.
  - Use of O-notation often makes it much easier to analyze algorithms; we can easily prove the $O(n^2)$ insertion-sort time bound by saying that both loops run in $O(n)$ time.
  - We often abuse the notation a little:
    * We often write $f(n) = O(g(n))$ instead of $f(n) \in O(g(n))$.
    * We often use $O(n)$ in equations: e.g. $2n^2 + 3n + 1 = 2n^2 + O(n)$ (meaning that $2n^2 + 3n + 1 = 2n^2 + f(n)$ where $f(n)$ is some function in $O(n)$).
    * We use $O(1)$ to denote constant time.

2.2 Ω-notation (big-Omega)

$\Omega(g(n)) = \{f(n) : \exists\ c, n_0 > 0$ such that $c|g(n)| \leq |f(n)|\ \forall n \geq n_0\}$

- $\Omega(\cdot)$ is used to asymptotically *lower bound* a function.

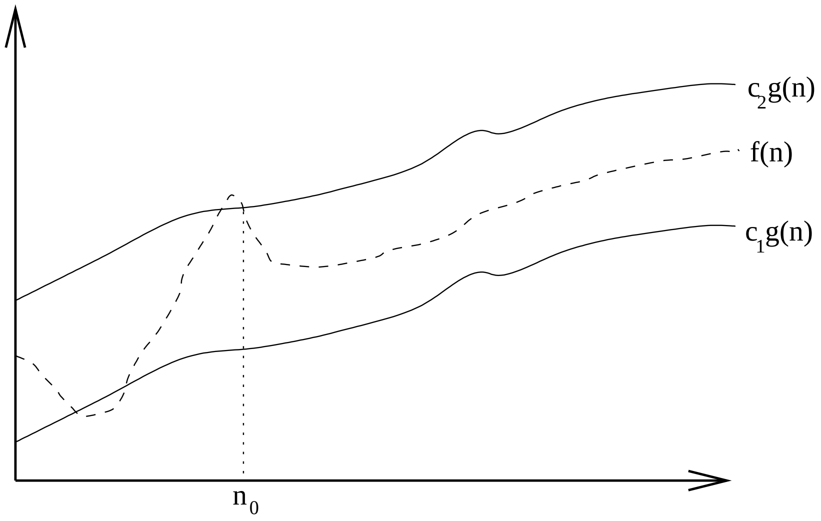


- Examples:
  - $1/3n^2 - 3n = \Omega(n^2)$ because $1/3n^2 - 3n \geq cn^2$ if $c \leq 1/3 - 3/n$ which is true if $c = 1/6$ and $n > 18$.
  - $k_1 n^2 + k_2 n + k_3 = \Omega(n^2)$.
  - $k_1 n^2 + k_2 n + k_3 = \Omega(n)$ (lower bound!)
- Note:
  - When we say "the running time is $\Omega(n^2)$", we mean that the *best case* running time is $\Omega(n^2)$ — the worst case might be worse.
- Insertion-sort:
  - Best case: $\Omega(n)$
  - Worst case: $O(n^2)$
  - We can also say that the *worst case* running time is $\Omega(n^2) \Longrightarrow$ worst case running time is "precisely" n^2.

2.3 Θ-notation (Big-Theta)

$\Theta(g(n)) = \{f(n) : \exists\ c_1, c_2, n_0 > 0$ such that $c_1|g(n)| \leq |f(n)| \leq c_2|g(n)|\ \forall n \geq n_0\}$

- $\Theta(\cdot)$ is used to asymptotically *tight bound* a function.



$$\boxed{f(n) = \Theta(g(n)) \text{ \textit{if and only if} } f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n))}$$

- Examples:
  - $k_1 n^2 + k_2 n + k_3 = \Theta(n^2)$
  - *worst case* running time of insertion-sort is $\Theta(n^2)$
  - $6n\log n + \sqrt{n}\log^2 n = \Theta(n \log n)$:
    * We need to find n_0, c_1, c_2 such that $c_1 n \log n \le 6n\log n + \sqrt{n}\log^2 n \le c_2 n \log n$ for $n > n_0$
      $c_1 n \log n \le 6n\log n + \sqrt{n}\log^2 n \implies c_1 \le 6 + \frac{\log n}{\sqrt{n}}$. Ok if we choose $c_1 = 6$ and $n_0 = 1$.
      $6n\log n + \sqrt{n}\log^2 n \le c_2 n \log n \implies 6 + \frac{\log n}{\sqrt{n}} \le c_2$. Is it ok to choose $c_2 = 7$? Yes, $\log n \le \sqrt{n}$ if $n \ge 2$.
    * So $c_1 = 6$, $c_2 = 7$ and $n_0 = 2$ works.

- Note:
  - We often think of $f(n) = O(g(n))$ as corresponding to $f(n) \le g(n)$.
  - Similarly, $f(n) = \Theta(g(n))$ corresponds to $f(n) = g(n)$
  - Similarly, $f(n) = \Omega(g(n))$ corresponds to $f(n) \ge g(n)$
  - One can also define o and ω
    * $f(n) = o(g(n))$ corresponds to $f(n) < g(n)$
    * $f(n) = \omega(g(n))$ corresponds to $f(n) > g(n)$

2.4 Asymptotic equality

$$\boxed{f(n) \sim g(n), \text{ as } n \to \infty, \text{ iff } \lim_{n\to\infty} \frac{f(n)}{g(n)} = 1}$$

- Strongest notion
- $f(n) \sim g(n) \quad \implies \quad f(n) = \Theta(g(n))$.

- *L'Hospital's Rule*: If $f(n)$ and $g(n)$ are differentiable and either $f(n)$ and $g(n) \to \infty$ or $f(n)$ and $g(n) \to 0$ as $n \to \infty$, then

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}.$$

- Example: As $n \to \infty$,

$$\begin{aligned}
\left(1+\frac{1}{n}\right)^n &= \exp\ln\left(1+\frac{1}{n}\right)^n && (1)\\
&= \exp\left(n\ln\left(1+\frac{1}{n}\right)\right) && (2)\\
&= \exp\left(\frac{\ln\left(1+\frac{1}{n}\right)}{1/n}\right) && (3)\\
&\to \exp\left(\frac{-(1/n^2)\Big/\left(1+\frac{1}{n}\right)}{-1/n^2}\right), \quad \text{by L'Hospital's Rule} && (4)\\
&= \exp\lim_{n\to\infty}\frac{1}{\left(1+\frac{1}{n}\right)} && (5)\\
&= \exp(1) = e. && (6)
\end{aligned}$$

2.5 Growth rate of standard functions

- Book introduces standard functions in section 2.2 (we will introduce them as we need them):
  - Polynomial of degree d: $p(n) = \sum_{i=1}^{d} a_i \cdot n^i$ where $a_1, a_2, \ldots, a_d$ are constants (and $a_d > 0$). $p(n) = \Theta(n^d)$
- "Growth order": $\log\log n, \log n, \sqrt{n}, n, n\log\log n, n\log n, n\log^2 n, n^2, n^3, 2^n$
  - Growth rate of polynomials versus exponentials: $\lim_{n\to\infty} \frac{n^b}{a^n} = 0$.

3 Summations

When analyzing insertion-sort we used an *arithmetic series*

$$\boxed{\sum_{k=1}^{n} k = 1+2+3+\cdots+n = \frac{n(n+1)}{2} = \Theta(n^2)}$$

How can we prove this?

- Asymptotic:

  Often good estimates can be found by using the largest value to bound others:

  $\sum_{k=1}^{n} k \leq \sum_{k=1}^{n} n = n \cdot \sum_{k=1}^{n} 1 = n^2 = O(n^2)$

  Another trick: Splitting the sum:

  $\sum_{k=1}^{n} k = \sum_{k=1}^{n/2-1} k + \sum_{\frac{n}{2}}^{n} k \geq \sum_{k=1}^{n/2-1} 0 + \sum_{\frac{n}{2}}^{n} k \geq (\frac{n}{2})^2 = \Omega(n^2)$.

  $\Downarrow$

  $\sum_{k=1}^{n} k = \Theta(n^2)$

- The *exact* answer can be gotten by method used by Gauss as a boy in grade school: Write the sum forwards, and immediately below it write the sum backwards, and then sum the n columns. Each of the n columns sums to $n+1$. Therefore, double the summation is $n(n+1)$. QED.

- Another way (**proof by induction!**):
  - Basis: $n = 1 \Longrightarrow$ $\sum_{k=1}^{1} = 1$
    $\frac{n(n+1)}{2} = \frac{1 \cdot 2}{2} = 1$
  - Induction:
    Assume it holds for n: $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$
    Show it holds for $n+1$: $\sum_{k=1}^{n+1} k = \frac{(n+1)(n+2)}{2} = \frac{1}{2}n^2 + \frac{3}{2}n + 1$
    Proof:

$$\begin{aligned}
\sum_{k=1}^{n+1} k &= \sum_{k=1}^{n} k + (n+1) \\
&= \frac{n(n+1)}{2} + (n+1) \\
&= \frac{1}{2}n^2 + \frac{1}{2}n + n + 1 \\
&= \frac{1}{2}n^2 + \frac{3}{2}n + 1
\end{aligned}$$

In general we can prove that $\boxed{\sum_{k=1}^{n} k^d = \Theta(n^{d+1})}$

Another important sum (*Geometric series*):

$$\boxed{\sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots x^n = \frac{x^{n+1} - 1}{x - 1} = O(x^n), \text{ for } x > 1}$$

- Can be derived by trivial identity

$$\sum_{k=0}^{n} x^k = 1 + x\Big(\sum_{k=0}^{n} x^k\Big) - x^{n+1}$$

.

- Proof by induction:
  - Basis: $n = 1 \Longrightarrow$ $\sum_{k=0}^{1} x^k = 1 + x$
    $\frac{x^{n+1}-1}{x-1} = \frac{x^2-1}{x-1} = \frac{(x+1)(x-1)}{(x-1)} = x + 1$
  - Induction:
    Assume holds for n: $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$
    Show it holds for $n+1$: $\sum_{k=0}^{n+1} x^k = \frac{x^{n+2}-1}{x-1}$
    Proof:

$$\begin{aligned}
\sum_{k=0}^{n+1} x^k &= \sum_{k=0}^{n} x^k + x^{n+1} \\
&= \frac{x^{n+1} - 1}{x - 1} + x^{n+1}
\end{aligned}$$

$$\begin{aligned} &= \frac{x^{n+1} - 1 + x^{n+1}(x-1)}{x-1} \\ &= \frac{x^{n+1} - 1 + x^{n+2} - x^{n+1}}{x-1} \\ &= \frac{x^{n+2} - 1}{x-1} \end{aligned}$$

- Asymptotic (we don't need to know result to do induction!):

  Consider for example that we want to prove that $\sum_{k=0}^{n} 3^k = O(3^k)$, that is, that $\sum_{k=0}^{n} 3^k \leq c3^n$ for some c.

  - Basis: $n = 1 \Longrightarrow \begin{array}{l} \sum_{k=0}^{1} 3^x = 1 + 3 = 4 \\ c3^1 = c3 \end{array}$

    Ok if $c > 4/3$
  - Induction:

    Assume holds for n: $\sum_{k=0}^{n} 3^k \leq c3^n$

    Show holds for $n+1$: $\sum_{k=0}^{n+1} 3^k \leq c3^{n+1}$

    Proof:

    $$\begin{aligned} \sum_{k=0}^{n+1} 3^k &= \sum_{k=0}^{n} 3^k + 3^{n+1} \\ &\leq c3^n + 3^{n+1} \\ &= c3^{n+1}(1/3 + 1/c) \\ &\leq c3^{n+1} \end{aligned}$$

    If $1/3 + 1/c < 1$ which holds if $c > 3/2$

Another important sum: $$\boxed{\begin{aligned} H_n &= \sum_{k=1}^{n} \frac{1}{k} \\ &= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \\ &= \ln n + \gamma + \frac{1}{2n} - \frac{1}{12n^2} + \ldots \end{aligned}}$$ (*Harmonic Series*)

where $\gamma \approx 0.44742\ldots$.

- Upper bound (approximate by superior integral, as in handout for sum of squares)

  $$\begin{aligned} \sum_{k=1}^{n} \frac{1}{k} &\leq 1 + \int_{1}^{n} \frac{1}{x}\,dx \\ &= 1 + \ln n - \ln 1 \\ &= 1 + \ln n \end{aligned}$$

- Lower bound (approximate by inferior integral)

  $$\sum_{k=1}^{n} \frac{1}{k} \geq \int_{1}^{n+1} \frac{1}{x}\,dx$$

$$
\begin{aligned}
&= \ln(n+1) - \ln 1 \\
&= \ln(n+1) \\
&= \ln\left(n\left(1+\frac{1}{n}\right)\right) \\
&= \ln n + \ln\left(1+\frac{1}{n}\right) \\
&= \ln n + \frac{1}{n} - \frac{1}{2n^2} + \ldots
\end{aligned}
$$

4 Growth review

- $O(\cdot)$ used to asymptotically upper bound functions.
- $\Omega(\cdot)$ used to asymptotically lower bound functions.
- $\Theta(\cdot)$ used to asymptotically tight bound functions.

5 Summation review

- We computed a number of sum's using:
  - Manipulation
  - Splitting and bounding terms ideas
  - Induction (!)
  - Approximation by an integral