CS 329e - Algorithms for Bioinformatics

Jan 29, 2008

Tandy Warnow

Topics for today

- Easy techniques for Big-oh analysis
- Homework assignment

Big-oh notation

- A function f(n) is O(g(n)) if there is some pair of constants c and c' such that

  $f(n) \leq cg(n)$ whenever $n > c'$

  This is not a statement of equality, but of an ***upper bound.***

  So n^2 is $O(n^3)$ but n^3 is not $O(n^2)$

- To determine if f(n) is O(g(n)), just look at the dominant terms - and ignore coefficients

Easy stuff

- Easy things:
  - n^k is $O(n^{k'})$ for all $k<k'$.
  - Every polynomial $f(n)$ is $O(c^n)$ as long as $c>1$.
  - $\log(n)$ is $O(f(n))$ when $f(n)$ is a polynomial

An “easy” proof technique

- To see if f(n) is O(g(n)) compute

  $\lim_{n\to\infty} f(n)/g(n)$

- If this limit exists, then f(n) is O(g(n)) *if and only if* $\lim_{n\to\infty} f(n)/g(n) \leq C$ (for some constant C)

- Examples:
  - $3n^2+5$ is $O(n^2)$
  - $87n^2+5000$ is $O(n^2)$
  - $5n^2$ is $O(n^3-100n)$
  - But n^3 is *not* $O(n^2)$

L'Hôpital's Rule

- If f(n) and g(n) are both differentiable and both go to infinity as $n \to \infty$, then

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

- Example: To determine if ln(n) is O(n), we compute $\lim_{n\to\infty} \ln(n)/n$.
- Since both ln(n) and n go to infinity, we use L'Hôpital's Rule, and obtain

  $$\lim_{n\to\infty} \ln(n)/n = \lim_{n\to\infty} (1/n)/1 = 0.$$

- Hence, ln(n) is O(n)

True/False Big-Oh statements

1. $87n^2+5000$ is $O(n^2)$
2. $5n^2$ is $O(n^3)$
3. n^3 is $O(n^2)$
4. n^3 is $O(5n^3)$
5. $\ln(n)$ is $O(n)$
6. $(\ln(n))^6$ is $O(n)$
7. n^3 is $O(n!)$
8. n^3 is $O(2^n)$
9. $n!$ is $O(2^{2n})$
10. 2^n is $O(n^n)$

Another technique: logarithms

- Suppose f(n) is O(g(n)). Then there are constants C, C' s.t. $f(n) \le Cg(n)$ for all $n > C'$.
- Hence $\ln(f(n)) \le \ln(Cg(n)) = \ln(C) + \ln(g(n))$ for all $n>C'$.
- Therefore, **$\ln(f(n))-\ln(g(n)) \le C^*$** (for some constant C^*)
- *Therefore, to check if f(n) is O(g), compute ln(f(n))-ln(g(n)) and see if this is bounded from above by a constant.*

Example: $f(n) = 2^n$ and $g(n)=n^2$

- To check if f(n) is O(g(n)), we compute logs:
  - $\ln(f(n)) = \ln(2^n) = n \ln(2)$
  - $\ln(g(n)) = \ln(n^2) = 2 \ln(n)$
- $\ln(f(n)) - \ln(g(n)) = n \ln(2) - 2 \ln(n)$, which is not bounded from above by a constant. So f(n) is not O(g(n)).
- On the other hand, g(n) is O(f(n)) since $\ln(g(n)) - \ln(f(n)) = 2 \ln(n) - n \ln(2)$ is bounded from above by a constant.