CS60020: Foundations of Algorithm Design and Machine Learning

Sourangshu Bhattacharya

ORDER NOTATION

Order Arithmetic

- Big-Oh notation provides a way to compare two functions
- "f(n) is **O**(g(n))" means:

  f(n) is less than or equal to g(n) up to a constant factor for large values of n

Categorizing functions

- Big-Oh can be used for categorizing or characterizing functions
- For example, the statements:
  $2n + 3$ is $\mathbf{O}(n)$ and $5n$ is $\mathbf{O}(n)$
  place $2n + 3$ and $5n$ in the same category
  - Both functions are less than or equal to $g(n) = n$, up to a constant factor, for large values of n
  - If the functions are running times of two algorithms, the algorithms are thus comparable

Definition of Big-Oh

f(n) is **O**(g(n)) if there is a real constant $c > 0$ and an integer constant $n_0 >= 1$ such that

$$f(n) <= c\, g(n), \text{ for } n >= n_0$$

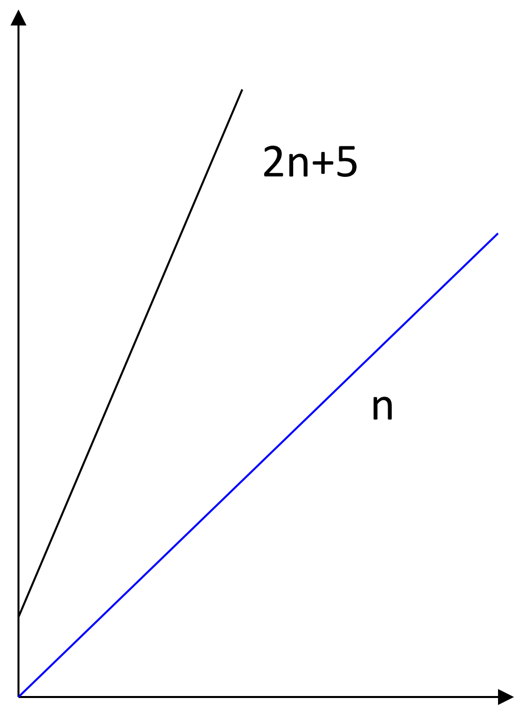
less than or equal up to a constant factor

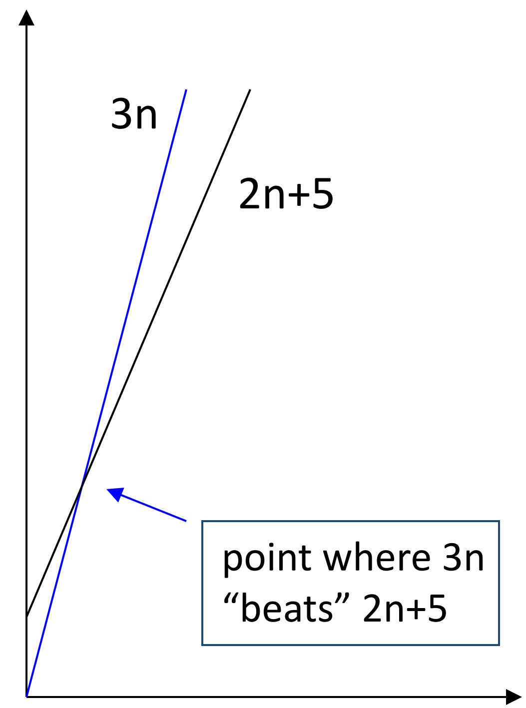
for large values of n

Example

- $f(n) = 2n + 5$
  $g(n) = n$
- Consider the condition
  $$2n + 5 <= n$$
  will this condition ever hold? No!
- How about if we tack a constant to n?
  $$2n + 5 <= 3n$$
  the condition holds for values of n greater than or equal to 5
- This means we can select $c = 3$ and $n_0 = 5$

Example





2n+5 is **O**(n)

Use of Big-Oh notation

- Big-Oh allows us to ignore constant factors and lower order (or less dominant) terms

$$2n^2 + 5n - 4 \text{ is } \mathbf{O}(\, n^2 \,)$$

constants

lower order terms

Function categories revisited

- The constant function: $f(n) = 1$
- The linear function: $f(n) = n$
- The quadratic function: $f(n) = n^2$
- The cubic function: $f(n) = n^3$
- The exponential function: $f(n) = 2^n$
- The logarithm function: $f(n) = \log n$
- The n log n function: $f(n) = n \log n$

Functions by increasing growth rate

- The constant function: $f(n) = 1$
- The logarithm function: $f(n) = \log n$
- The linear function: $f(n) = n$
- The n log n function: $f(n) = n \log n$
- The quadratic function: $f(n) = n^2$
- The cubic function: $f(n) = n^3$
- The exponential function: $f(n) = 2^n$

Big-Oh as an upper bound

- The statement f(n) is **O**(g(n)) indicates that g(n) is an **upper bound** for f(n)
- Which means it is also correct to make statements like:
  - 3n+5 is **O**(n^2)
  - 3n+5 is **O**(2^n)
  - 3n+5 is **O**(5n + log n - 2)
  - But the statement 3n+5 is **O**(n) is the “tightest” statement one can make

Relatives of Big-Oh

- Big Omega Ω : lower bound
- Big Theta Θ : the function is both a lower bound and an upper bound
- For this course, only Big-Oh notation will be used for algorithm analysis