- Idea: roughly measure the running time/memory for algorithm

\[ 3n^2 + 2n \approx n^2 \neq 2^n \]

- Definition

\[
\begin{align*}
\leq^* & \quad (1) \quad f(n) = O(g(n)), \text{ if there is constants } C > 0, \ n_0 > 0 \text{ such that for all } n \geq n_0, \ f(n) \leq C \cdot g(n) \\
\geq^* & \quad (2) \quad f(n) = \Omega(g(n)), \text{ if there is constants } C > 0, \ n_0 > 0 \text{ such that for all } n \geq n_0, \ f(n) \geq C \cdot g(n) \\
=^* & \quad (3) \quad f(n) = \Theta(g(n)) \text{ if } f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n)) \\
<^* & \quad (4) \quad f(n) = o(g(n)) \text{ if } g(n) \neq O(f(n)) \\
>^* & \quad (5) \quad f(n) = \omega(g(n)) \text{ if } f(n) \neq O(g(n))
\end{align*}
\]

- Example: (a) \( 3n^2 + 2n = O(n^2) \)

Proof: \( 3n^2 + 2n \leq 3n^2 + 2n^2 = 5n^2 \)

in the definition, can choose \( C = 5, \ n_0 = 1 \)

\[ 3n^2 + 2n \leq C \cdot n^2, \text{ so } 3n^2 + 2n = O(n^2) \]

(b) \( n^2 \neq O(n) \)

\( (n^2 = \omega(n)) \)

Proof: Assume \( n^2 = O(n) \) (towards contradiction)

there is \( C > 0, \ n_0 > 0 \) s.t.

when \( n \geq n_0, \ n^2 \leq C \cdot n \)

want: find a \( n \geq n_0 \) s.t. \( n^2 > C \cdot n \)

\[ n^2 > C \cdot n \]

\[ \Rightarrow n > C \]

pick \( n > \max\{n_0, C\} \)

then we have \( n^2 = n \cdot n > C \cdot n \) contradiction!

therefore \( n^2 \neq O(n) \)

\[ \log n < \ln n < n < n \log n < n^2 < 2^n < 3^n < n! \]
- **Bubble Sort**

  \[ \text{for } i = n \text{ down to 1} \]
  \[ \quad \text{for } j = 1 \text{ to } i-1 \]
  \[ \quad \text{if } a[j] > a[j+1] \text{ then swap.} \]

  \[ T = (n-1) + (n-2) + \ldots + 1 = \frac{n(n-1)}{2} \]

- What if there is an algorithm that calls Bubble Sort on arrays of size 1, 2, 3, \ldots, n

  \[ T = \frac{1 \times 0}{2} + \frac{2 \times 1}{2} + \frac{3 \times 2}{2} + \ldots + \frac{n(n-1)}{2} \]

  \[ = \frac{2 \times 1 \times 0 - 1 \times 0 \times (n-1)}{6} + \frac{3 \times 2 \times 1 - 2 \times 1 \times 0}{6} + \frac{4 \times 3 \times 2 - 3 \times 2 \times 1}{6} \]

  \[ + \ldots + \frac{(n+1)n(n-1) - n(n-1)(n-2)}{6} \]

  \[ = \frac{(n+1)n(n-1)}{6} \]

  Claim: \[ T = \Theta(n^3) \]

  Proof: \[ T = \sum_{i=1}^{n} \frac{i(i-1)}{2} \leq n \cdot \frac{n(n-1)}{2} = O(n^3) \]

  \[ T = \sum_{i=1}^{n} \frac{i(i-1)}{2} > \sum_{i=n/2}^{n} \frac{i(i-1)}{2} \geq \frac{n^2}{2} \cdot \frac{1}{2} = \frac{n^3}{16} \]

  \[ T(n) = \Omega(n^3) \]

- **Euclid's algorithm**

- Goal: compute greatest common divisor (gcd) of 2 integers.

  \[ \text{gcd}(15, 9) = 3 \]

- gcd(a, b)

  \[ \text{if } b = 0 \text{ then} \]

  \[ \text{return } a \]

  \[ \text{else return } \text{gcd}(b, a \mod b) \]

  \[ \text{gcd}(15, 9) \rightarrow \text{gcd}(9, 6) \rightarrow \text{gcd}(6, 3) \rightarrow \text{gcd}(3, 0) \]

  \[ 3 \]

  Proof: By induction

  \[ 0 \text{ if } b = 0 \quad \text{gcd}(a, 0) = a. \]
Induction hypothesis (assume my alg works on small inputs): Assume \( \gcd(a, b) \) is correct when \( b < n \) (know this is true for \( n = 1 \)).

Want to prove \( \gcd(a, b) \) is correct when \( b = n \) (my alg also works for larger inputs).

Want: \( \gcd(a, b) = \gcd(b, a \mod b) \)

Proof:

Assume \( a \mod b = a - k \cdot b \)

1. If \( c \mid a, c \mid b \) then \( c \mid a \mod b \)

   - \( c \) divides \( a \)
   
   \[
   a \mod b = \frac{a - kb}{c} = \frac{a}{c} - \frac{kb}{c} = \frac{a}{c} - 1 \cdot \frac{b}{c} = \text{integer}
   \]

2. If \( c \mid b \) then \( c \mid a \mod b \)

   \[
   \frac{a}{c} = \frac{(a - kb) + kb}{c} = \frac{a - kb}{c} + \frac{kb}{c} = \text{integer}
   \]

(1) + (2) \implies the set of common divisors for \( (a, b) \) and \( (b, a \mod b) \) are the same.

\[
\gcd(a, b) = \gcd(b, a \mod b)
\]

By induction hypothesis, since \( a \mod b < b = n \)

\( \gcd(b, a \mod b) \) is computed correctly.

Therefore, \( \gcd(a, b) \) is also correct.