\[ f(n) = \omega(g(n)) : \]

\[ \forall c > 0, \exists N, \forall n \geq N \]

\[ f(n) > c \cdot g(n) \]

\[ (\iff g(n) = o(f(n)) ) \]

\[ (\iff \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty ) \]

**Myths:**

*If* \( f(n) \neq \Omega(g(n)) \) *then*

\[ f(n) = o(g(n)) \quad \text{FALSE} \]

\[ \exists g \]

\[ f(n) \neq O(g(n)) \]

\[ g(n) \neq O(f(n)) \]

**Exponentials**

\[ a > 1 \text{ fixed real } \quad b \in \mathbb{R} \]

\[ \lim_{n \to \infty} \frac{n^b}{a^n} = 0. \]

\[ n^b = o(a^n) \]

\[ \log_b a = \frac{1}{\log_a b} \]

Starting Algos
Problems

types:
  decision
  search
  arrangement
  optimization

Problem instance — input to algo

Analyze algos for correctness
  efficiency
  space

Model of computation is required.

RAM model
  (sequential algos)
  random access machine

\[
\begin{array}{ccccccc}
R_0 & R_1 & R_2 & \ldots \\
\vdots & \vdots & \vdots & \ddots
\end{array}
\]

Each register \( R_i \) can hold an integer.

Fixed instruction set including, e.g.

\[ R[i] := R[j] + R[k] \]

(for any constant \( i, j, k \geq 0 \))

\[ R[i] := -R[j] \]

\[ R[i] := R[j] \]

\[ R[i] := R[j] \ast R[k] \]

\[ R[i] := R[j] \div R[k] \]

\[ R[i] := R[j] \mod R[k] \]

Could allow register contents to be real (floating point)

\[ R[i] := R[j] / R[k] \]

(real division)

To keep multiplication and "primitive" (constant time), restrict register values to \( O(1 \lg n) \) bits, where \( n \) is the input size.

\[ R[i] := R[R[j]] \]

\[ R[R[i]] := R[j] \]

If values are \( O(1 \lg n) \) bits then ok to call these primitive.
If \( R[i] \bowtie R[j] \) then
\[
\text{goto } L
\]
\[
\leq \Rightarrow \Rightarrow
\]
\[
\text{Goto } L
\]
\[
R[i] := k
\]
\[
\text{Read } R[i]
\]
\[
\text{Write } R[i]
\]

Each instruction execution takes unit time. (constant time)

\[
\rightarrow c \log n \text{ bits}
\]
\[
\Rightarrow \text{can pick } c \text{ large enough.}
\]

Data structures

\[
\text{Algos} \rightarrow \text{Data Structs}
\]

Sorting

\[
\text{Insertion, Selection, Mergesort, Quicksort, Heapsort}
\]

\[
O(n^2)
\]

\[
\text{sorted}
\]
\( O(n^2) \) \quad \text{worst-case time} \\

\[
\sum_{i=2}^{n} i = \sum_{i=1}^{n} i - 1
\]
\[
= \frac{n(n+1)}{2} - 1
\]
\[
= \Theta(n^2)
\]

\[
\left| \begin{array}{c}
|a_i| \\
1
\end{array} \right|
\]

\( a_1 < a_2 < \ldots < a_n \)

\text{Best case} = \Theta(n)

\text{Selection:} \quad \Theta(n^2)

\text{worst case:} \quad \Theta(n^2)

\text{best case:}
\[ f(n) = O(g(n)) \]
\[ \exists c > 0, \exists N, \forall n \geq N, f(n) \leq c \cdot g(n) \]
\[ f(n) = \Omega(g(n)) \]
\[ \exists c > 0, \exists N, \forall n \geq N, f(n) \geq c \cdot g(n) \]
\[ f(n) = \Theta(g(n)) \]

means:
\[ f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n)) \]

\[ f(n) = \Omega(g(n)) \quad \iff \quad g(n) = O(f(n)) \]
\[ f(n) = o(g(n)) \]

\[ \forall c > 0 \exists N \forall n \geq N, f(n) < c \cdot g(n) \]
\[ \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0. \]

Generally assume: \( f, g \) are \underline{asymptotically nonnegative}.

\[ \exists N, \forall n \geq N, f(n) \geq 0, \quad g(n) \geq 0. \]
\[ f(n) = \omega(g(n)). \]