Asymptotic Analysis

High-level idea: suppress constant factors and lower order terms.

Example: merge-sort $6N \cdot \log_2 N + 6N$ simplify to $N \cdot \log_2 N$.

Terminology: Running time is $O(N \cdot \log_2 N)$.

N = input size (e.g., array length).

Big Oh Let $T(N)$ = function on $N = 1, 2, 3, \ldots$

When $T(N) = O(f(N))$?

If for all sufficiently large values of $N$, $T(N)$ is bounded above by a constant multiple of $f(N)$. 
**Formal definition**

\[ T(N) = O\left(f(N)\right) \]

if and only if there exists constants \( C, N_0 > 0 \) such that

\[ T(N) \leq C \cdot f(N) \]

for all \( N \geq N_0 \)

**Warning**: \( C, N_0 \) cannot depend on \( N \).

---

**Omega Notation**

Definition: \( T(N) = \Omega(f(N)) \) if and only if \( \exists \) constants \( C, N_0 \) such that

\[ T(N) \geq C \cdot f(N) \]

for all \( N \geq N_0 \)

---

**Theta Notation**

Definition: \( T(N) = \Theta(f(N)) \) if and only if

\[ T(N) = O(f(N)) \quad \text{and} \quad T(N) = \Omega(f(N)) \]

Equivalently \( \exists \) constants \( c_1, c_2 \) and \( N_0 \) such that

\[ c_1 f(N) \leq T(N) \leq c_2 f(N) \]

for all \( N \geq N_0 \)
Claim: \(2^{n+10} = O(2^n)\)

Proof: We need to pick 2 constants \(c, N_0\) such that the inequality holds for all \(N \geq N_0\):

\[ 2^{n+10} \leq c \cdot 2^N \]

\[ 2^{n+10} = 2^{10} \cdot 2^n = 1024 \cdot 2^n \]

so if choose \(c = 1024\), \(N_0 = 1\) then the inequality holds for all \(N \geq N_0\). \(\boxed{\text{QED}}\)

Claim: \(2^n\) is not \(O(2^N)\)

Proof by contradiction. If \(2^n = O(2^N)\) then there exist constants \(c, N_0 > 0\) such that:

\[ 2^n \leq c \cdot 2^N \quad \forall N \geq N_0 \]

But then (cancelling \(2^n\)):

\[ 1 \leq c \quad \forall N \geq N_0 \]

But this inequality is false since \(c\) is a fixed constant and \(N\) can go to \(\infty\). Therefore \(2^n\) will surpass \(c\). \(\boxed{\text{QED}}\)