1 Asymptotic Analysis

1.1 Big-Oh Notation

Concerns functions defined in positive integers

Let $T(n)$ =Function on $n = 1, 2, 3, \ldots$

[usually, the worst-case running time of an algorithm]

Q: When is $T(n) = O(f(n))$?

A: if eventually for all sufficiently large n, $T(n)$ is bounded above by a constant multiple of $f(n)$.

Formal Definition:

$$T(n) = O(f(n)) \iff \exists_{c,\, n_0} > 0 | \forall_{n \geq n_0} T(n) \leq cf(n)$$

c, n_0 constants independent of n.

1.2 Basic Examples

Claim: If $T(n) = a_k n^k + \ldots + a_1 n + a_0$ then $T(n) = O(n^k)$. This says big O notation really supresses the constant terms.

Proof:

Choose:

$n_{0=1}$

$c = |a_k| + |a_{k-1}| + \ldots + |a_1| + |a_0|$

We need to show that $\forall_{n \geq 1}$, $T(n) \leq cn^k$

We have,

$$\begin{aligned} \forall_{n \geq 1},\, T(n) &\leq |a_k| n^k + \ldots + |a_1| n + |a_0| \\ &\leq |a_k| n^k + \ldots + |a_1| n^k + |a_0| n^k \\ &= cn^k \end{aligned}$$

How did we know c and n_0? Reverse engineer constants that work. Go through the proof with a generic values and then go back with working constants.

Claim: For every $k \geq 1$, $n^k$is not $O(n^{k-1})$

Proof: by contradiction

Suppose n^k was $O(n^{k-1})$. From definition, $\exists_{c,\, n_0 > 0} | n^k \leq cn^{k-1} \forall_{n \geq n_0}$

Cancel $k - 1$ from both sides of the inequality, so

$$\forall_{n \geq n_0},\, n \leq c$$

which is false, integers above$c$, e.g.$c + 1$, are not bounded by c.

1.3 Big Omega and Theta

Ω Definition:

$$T(n) = \Omega(f(n)) \iff \exists_{c,\, n_0} \mid \forall_{n \geq n_0} T(n) \geq cf(n)$$

Bound from below

Θ Definition:

$$T(n) = \Theta(f(n)) \iff T(n) = O(f(n)) \text{ and } \text{T(n)=}\Omega$$

Eventuall $T(n)$ is sandwitched between two different constant multiples of $f(n)$.

$$\exists_{c_1,\, c_2,\, n_0} \mid \forall_{n \geq n_0} c_1 f(n) \leq T(n) \leq c_2 f(n)$$

Generally we care about the uppoer bound and just say an algorithm runs in $O(n)$ time.

Let $T(n) = \frac{1}{2}n^2 + 3n$, then:

$T(n) = \Omega(n)$ to prove use e.g. $[n_0 = 1,\, c = \frac{1}{2}]$

$T(n) = \Theta(n^2)$ to prove use e.g. $[n_0 = 1,\, c_1 = \frac{1}{2},\, c_2 = 4]$

$T(n) = O(n^3)$ to prove use e.g. $[n_0 = 1,\, c = 4]$

1.3.1 Little-Oh Notation

Strictly less-than notation.

o Definition:

$$T(n) = o(f(n)) \iff \forall_{c>0} \exists_{n_0} \mid \forall_{n \geq n_0} T(n) \leq c f(n)$$