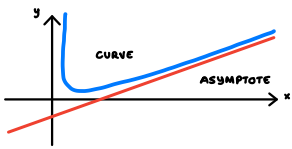


ASYMPTOTIC BOUNDING

Every algorithm requires resources \rightarrow TIME
 \rightarrow SPACE

We use asymptotic bounding to know how many resources we are using.

ASYMPTOTIC:



We have a function $T(n)$, which is the time/space the algorithm takes, as n changes.

We want to bound the change \rightarrow how does it change as it gets larger?

BIG O NOTATION

UPPER BOUND \rightarrow WORST CASE $\rightarrow T(n)$ is taking as much resources as it can

$T(n)$ is $O(f(n))$ iff $T(n) \leq C f(n)$ for all $n \geq n_0$.

We want $T(n)$ to stay below $f(n)$

We look for a constant C that will make this true

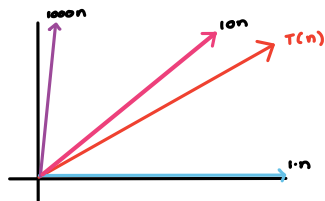
Let's try with: $f(n) = n$

$C f(n) = C \cdot n$

$C = 1 \rightarrow 1 \cdot n$

$C = 1000 \rightarrow 1000 \cdot n$

$C = 10 \rightarrow 10 \cdot n$



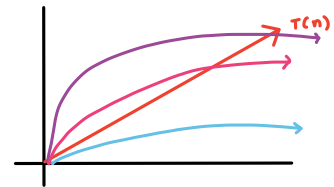
Let's try with: $f(n) = \log n$

$C f(n) = C \log(n)$

$C = 1 \rightarrow 1 \cdot \log(n)$

$C = 1000 \rightarrow 1000 \cdot \log(n)$

$C = 10 \rightarrow 10 \cdot \log(n)$



BIG OMEGA NOTATION

LOWER BOUND \rightarrow BEST CASE

BIG THETA NOTATION

EXACT BOUND