Alg 1

input: length n Array A of ints

if A[1] > 0:

    total = 0

    for i in 1, 2, ..., n:

        total = total + A[i]   ] $\Theta(n)$

else:

    total = 5

steps taken by Alg 1

on any input w/ A[1] ≤ 0: $T(n) = \Theta(1)$

on any input w/ A[1] > 0: $T(n) = \Theta(n)$

$\Omega(n)$ T or F

worst-case: $\Theta(n)$

$O(n^2)$ T or F ?

→ $O(n)$ T or F ?

best-case: $\Theta(1)$

→ $O(n)$ T or F $\Omega(n \log n)$ F

$\rightarrow O(n^2)$ T or F

My bank account has no more than $1 million.

worst-case runtime:

max # of steps needed on input of size n

best-case runtime:

min # of steps needed on input of size n.

$\sqrt{n} = O(\log n)$? no

$\log n = O(\sqrt{n})$? yes

Is $3^n = O(2^n)$? no

$3n^2 = O(n^2)$ true

Claim 3^n is not $O(2^n)$.

proof We need to show that there do not exist $n_0 \geq 0$, $c > 0$ such that

$$\forall n \geq n_0 : \quad 3^n \leq c\, 2^n .$$

To do so, we show how to construct an n such that

$$3^n > c\, 2^n$$

for any choice of c.

Consider the first n such that $(3/2)^n > c$.

Since $\left(\frac{3}{2}\right)^n \to \infty$ as $n \to \infty$, such an n must exist.

Now notice: $\left(\frac{3}{2}\right)^n > c$.

$$\frac{3^n}{2^n} > c$$

$3^n \supset \subset 2^n$

as needed.