Asymptotic analysis

Algorithm A

$T_A(n)$ = worst case running time

= for all inputs of length n, the largest numbers of lines of code executed on an input

$T(n) = O(n)$ "$T(n)$ is big-Oh of n"
(is → is equal to)

really means

$\forall n$ (for all suff. large) $T(n) \leq cn$

↑ constant

Consider a linked list A. How long to search for an element in A? $T(n)$ ← (how long to search)

n → length of linked list

(R) $O(1)$ ✓ (G) $O(n)$ (B) Neither are correct

```
while(curr != NULL)
{
    if (curr.data == val)
        return TRUE
    else
        curr = curr -> next
```

Upper bound

"$T(n) \leq n$"

Is $T(n) = \Omega(n)$?   (R) Y   (G) N

Lower bound ("$T(n) \geq n$")

If $\lim_{n \to \infty} \frac{T(n)}{n} > 0$, $T(n) = \Omega(n)$

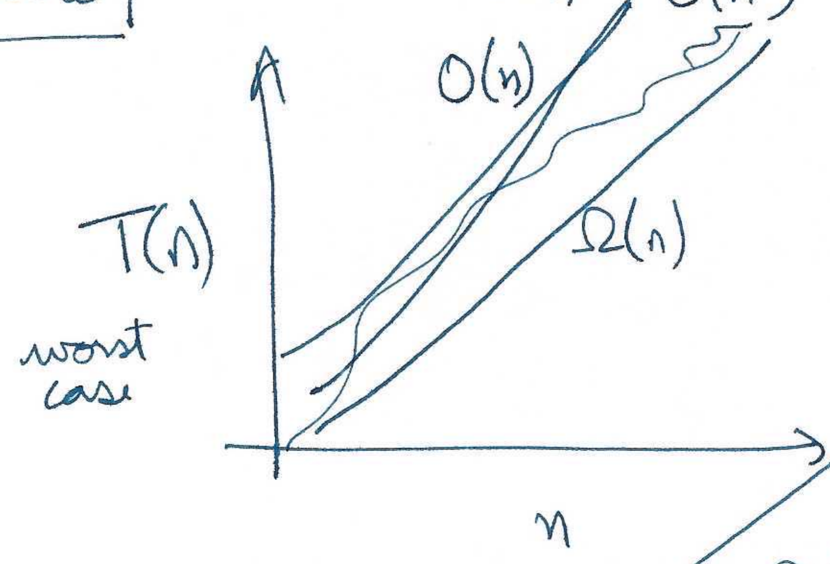
Ω does NOT mean best case.

In the worst case, there IS a line of code that run at least n times.

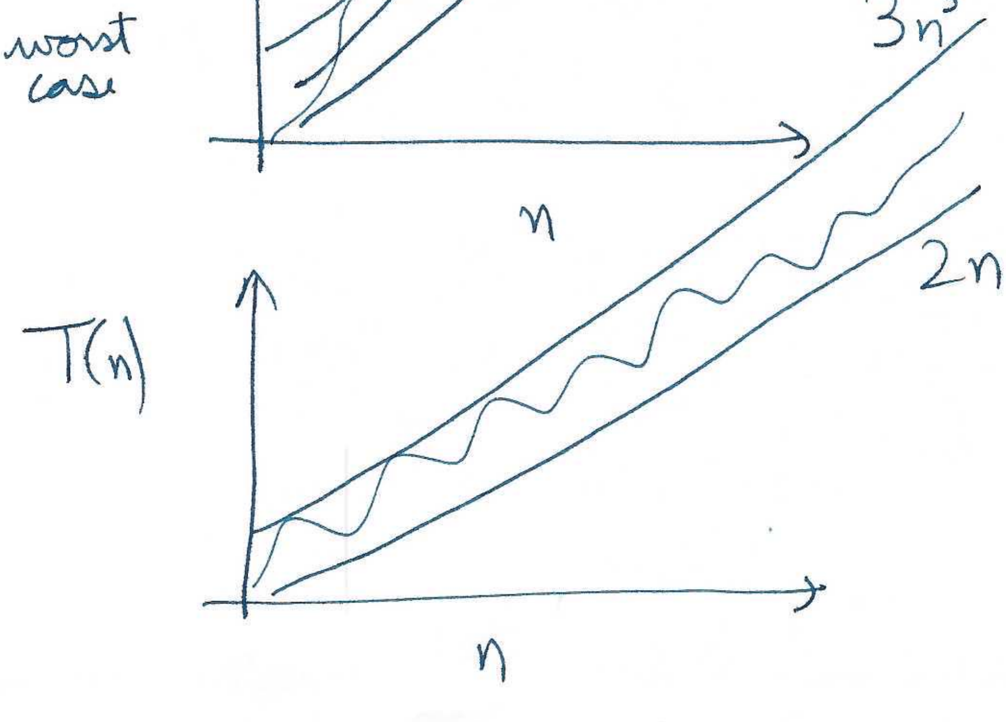
$T(n) = \Theta(n)$
Linear time

Tight bound ("$T(n) = n$")

$T(n) = \Theta(n^2)$



Correct $\begin{cases} T(n) = \Omega(n) \\ T(n) = O(n^2) \end{cases}$



For searching in a linked list, all lines of code run at most n times. Therefore, search time is $O(n)$.

~~There is~~ Suppose one searches for the last element in a linked list. There is a line of code (comparison in while condition/update of curr) that runs at least n times. Hence, search time is $\Omega(n)$.

What is running time of insert into linked list?

✓ ~~(A)~~ $O(1)$ (B) $O(n)$ (B) Something else

$T(n) = \Theta(1)$

Constant time → Time is independent of size of input.

$T(n)$ = worst case running time of selection sort

$T(n) = \Theta$(worst-case # of comparisons that selection sort makes)

How many comparisons?

i	# comparisons in inner loop
0	n
1	$n-1$
2	$n-2$
⋮	⋮
$n-1$	⋮

Total # comparisons $= n + (n-1) + (n-2) + \ldots 1$

$$= \sum_{i=0}^{n-1} (n-i) = \sum_{j=1}^{n} j = \frac{n(n+1)}{2} = O(n^2)$$

Sum of Arithmetic Progression

$$\frac{n(n+1)}{2} = \Omega(n^2)$$

$$\frac{n(n+1)}{2} = \Theta(n^2)$$

$T(n) = \Theta(n^2)$

How many comparisons?

i	# comp. in inner loop
0	0
1	1
2	2
⋮	⋮
	$\leq i$
⋮	⋮
$n-1$	$n-1$

upper bound

Worst case

$$\leq \text{Total \# comparisons} \leq \sum_{i=0}^{n-1} i = \frac{n(n-1)}{2} = O(n^2)$$

~~Ω~~

All inputs: We can conclude that Insertion sort runs in $O(n^2)$.

Find a bad input: Is the worst case running time of Insertion Sort $\Omega(n^2)$?

n	n-1	n-2	—	—

Suppose input was sorted.

0	1	2	3	-	-

How many comparisons does insertion sort make?

$\leq n$ comparisons

When input is ~~reverse of a so~~ in decreasing order, the # comparisons in i^{th} loop is at least i.

$$\text{Total \# comparison for WORST INPUT} \geq \sum_{i=0}^{n-1} i = \Omega(n^2)$$