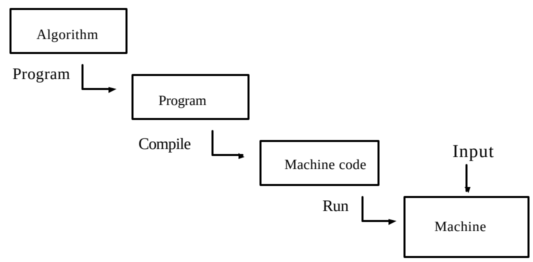
Is the algorithm efficient?

Resources to measure

- Time
- Space
- Random bits
- Etc.

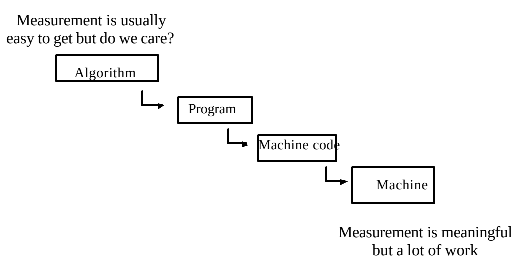
CS140 3-1

Time: Where to measure?



CS140 3-2

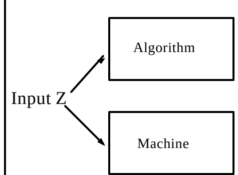
Where to measure?



CS140 3-3

A useful assumption

T_A and T_M differ by no more than a multiplicative constant



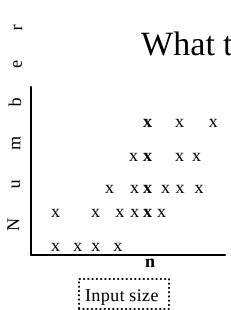
$T_A(Z)$: number of steps of algorithm on input Z

$$T_A(Z)/c < T_M(Z) < cT_A(Z)$$

$T_M(Z)$: number of steps of machine on input Z

CS140 3-4

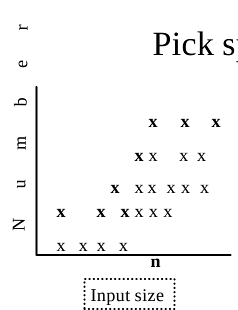
Time: What to measure?



- Run time depends on input size
- Run time can vary on different inputs of size n.

CS140 3-5

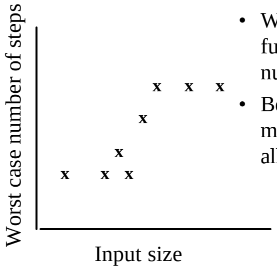
Pick special case



- Run time depends on input size
- Run time can vary on different inputs of size n.
- Choose case:
  - Worst case (show in bold)
  - Best case
  - Average case
  - Etc.

CS140 3-6

Worst case performance of algorithm A



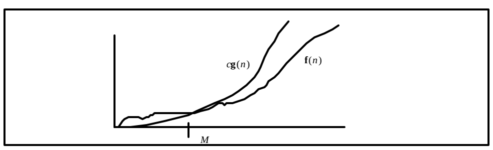
- We can compute this function at a finite number of points .
- Better yet, we can model this function for all input sizes.

A general problem …

- Question: How can we give a succinct description of an arbitrary function?
- Answer: Big-O notation.

Upper Bounds

- **f**:N→N and **g**:N→N are positive-valued, monotonically increasing functions.
- O(**g**(n)) = {**f**(n): there are constants c and M such that **f**(n)$\leq c$ **g**(n) for all $n \geq M$}



We may also say

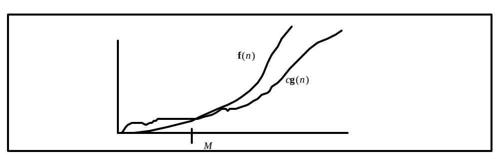
- **f**(n) = O(**g**(n)) **to mean** **f**(n) ε O(**g**(n))
- **f**(n) = n^2 + O(**g**(n)) **to mean** **f**(n)=n^2 +**h**(n) for some **h**(n) ε O(**g**(n))

Properties of upper bounds

- Big-O gives a partial order:
  Reflexive, transitive, but not symmetric
- A good test:
  If the sequence **f**(n)/**g**(n) n=1,2,… converges (i.e. to a constant c) then **f**(n)=O(**g**(n)).

Lower Bounds

Ω(**g**(n)) = {**f**(n): there are constants c and M such that **f**(n)$\geq c$**g**(n) for all $n \geq M$}



Properties of bounds

- Big-O and Ω give a partial order:
  Reflexive, transitive, *transpose symmetry*
- Transpose symmetry:
  f(n) = O(**g**(n)) ⬌ **g**(n) = Ω(**f**(n))

CS140 3-13

Other Bounds

Θ(**g**(n)) = {**f**(n): **f**(n) ε O(**g**(n)) and
f(n) ε Ω(**g**(n))}

o(**g**(n)) = {**f**(n): the sequence **f**(n)/**g**(n) diverges}

ω(**g**(n)) = {**f**(n): the sequence **g**(n)/**f**(n) diverges}

CS140 3-14

Summary

Partial Orders ⬌	Function Bounds
$\lesssim$	O
$\gtrsim$	Ω
=	Q
<	o
>	ω

CS140 3-15

Run time bounds for algorithm ▲

The running time of ▲ is O(n^3).

⬍

The worst case running time of ▲ is O(n^3).

⬍

▲ is O(n^3).

CS140 3-16