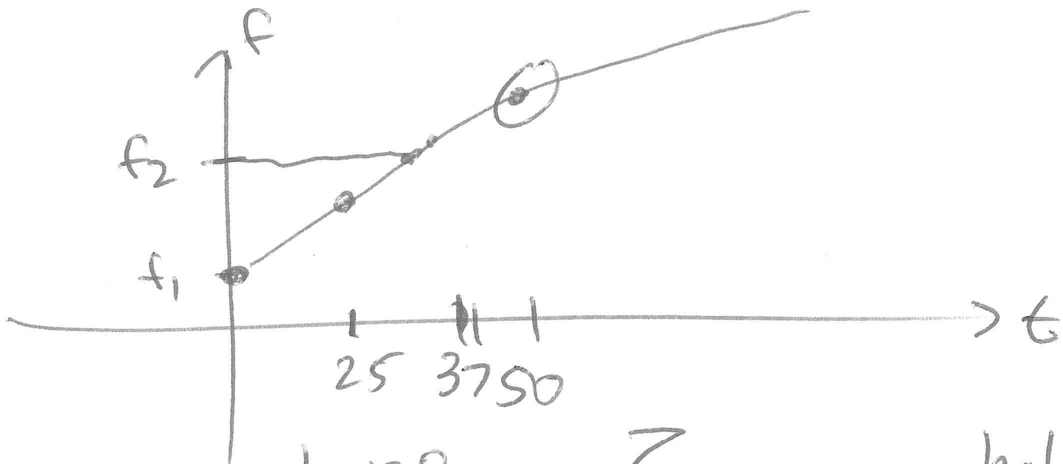
6/18/24

Binary Search / Bisection

$$\frac{f_2 - f_1}{f_2 f_1} = \underline{at} + b\,e^{-ct} \quad (a, b, c \text{ given})$$

(target → f_2, cur → f_1; at = linear, e^{-ct} = exponential)

D a function of t ↑ easy calc forward but difficult to do backwards.



low = 0 } guess halfway
high = $\frac{D}{a}$ } either low = mid
mid = (low+high)/2   high = mid

Hallmarks for binary search:

(a) easy calc "forwards", but hard to do backwards

(b) easy forward function is either increasing or decreasing (no turning pts)

Setting low + high at first

(1) real ans is such that
low ≤ real ans ≤ high

(2) range can't be so big that it causes math/overflow errors in the function you plug into!

Two types

(a) real valued ones
(b) integer ones

fixed # of iterations
50 → 100
mid = (low+high)/2.0;

while (low < high){
// (low+high+1)/2
int mid = (low+high)/2;

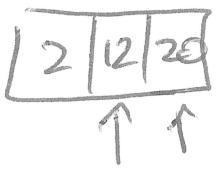
if (canDo(mid))
high = mid; // or mid-1

else
low = mid+1; // or mid

}

Test case
low=2, high=3

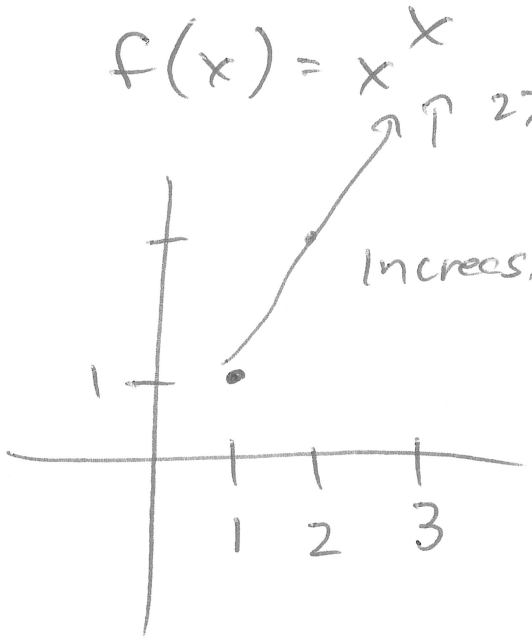
reg bin search



low ⌀ 2
high ⌀ 1
val 13
mid ⌀ 2

low = mid+1
high = mid-1

$f(x) = x^x$



$10^{10} = 10,000,000,000$

numbers have to move if:

(a) on diff rows

OR (b) if everything btw doesn't move.

What if I ask the easier question:
Given that my weightlifting ability is W, can I fix the weights?

* "Erase" all # $\leq W$, then check if in order

Trans