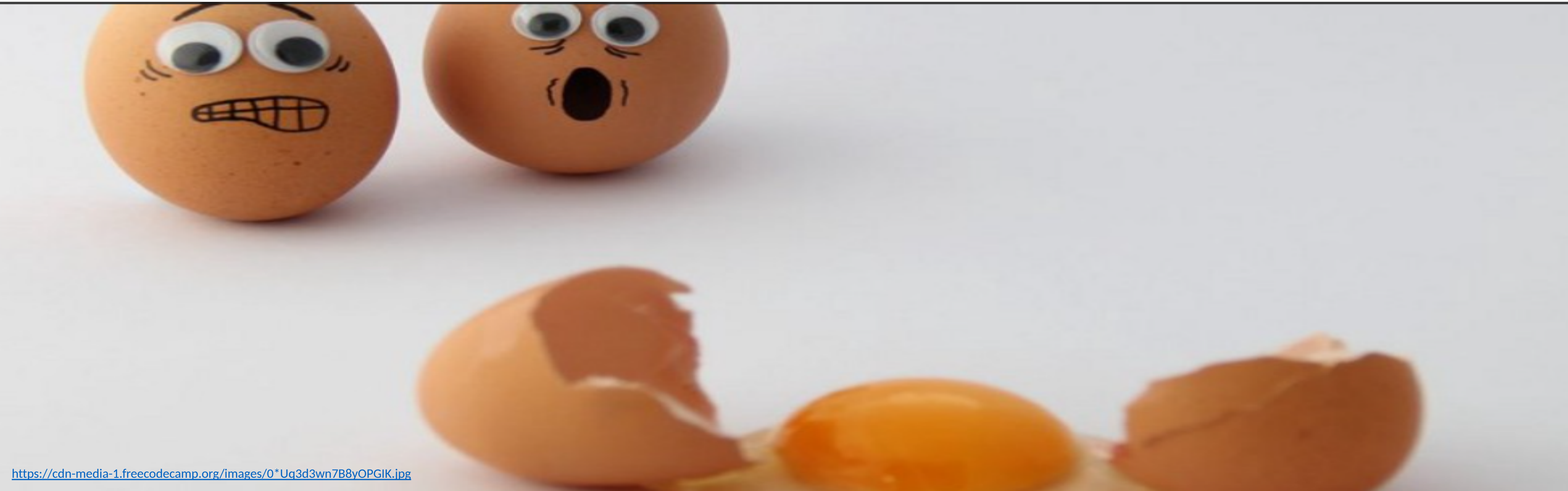
Egg Drop Mathematics:
It IS all its cracked up to be!

Steven J Miller, Williams College (sjm1@williams.edu) https://web.williams.edu/Mathematics/sjmiller/public_html/



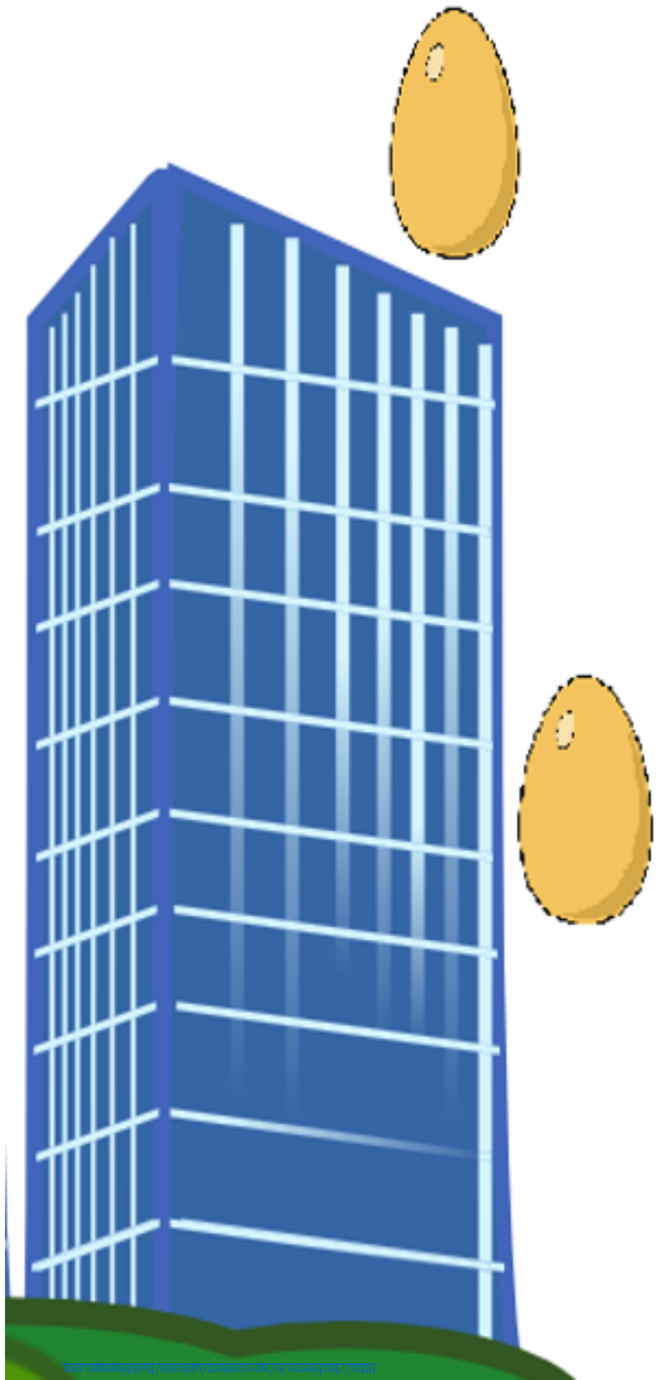
https://cdn-media-1.freecodecamp.org/images/0*Uq3d3wn7B8yOPGIK.jpg

Building with N floors, have 2 golden eggs.

Special eggs: some floor n such that if you drop from below n no damage; can drop as many times as wish.

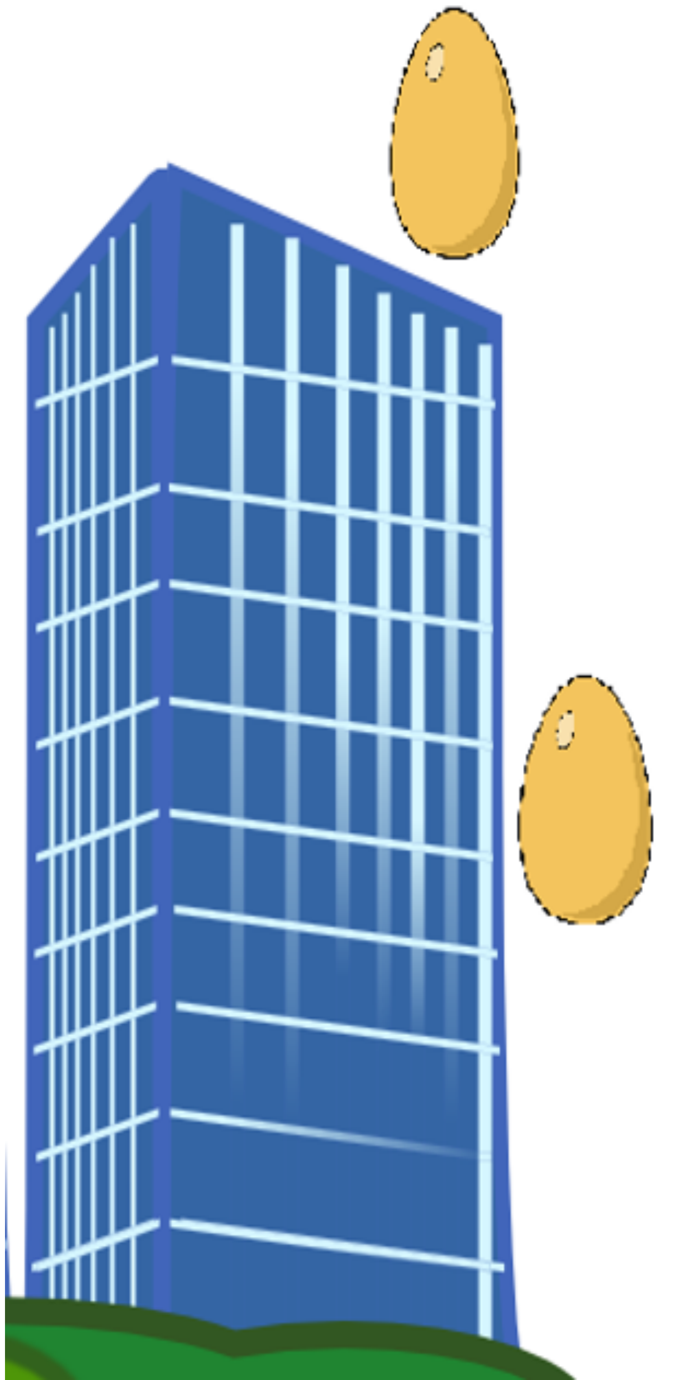
If drop even once from floor n or higher immediately break.

Find in as few drops as you can what n is (the lowest floor where if you drop from there it breaks). Doesn't matter if have any of the golden eggs at the end - just want to know n.



Interpretation:

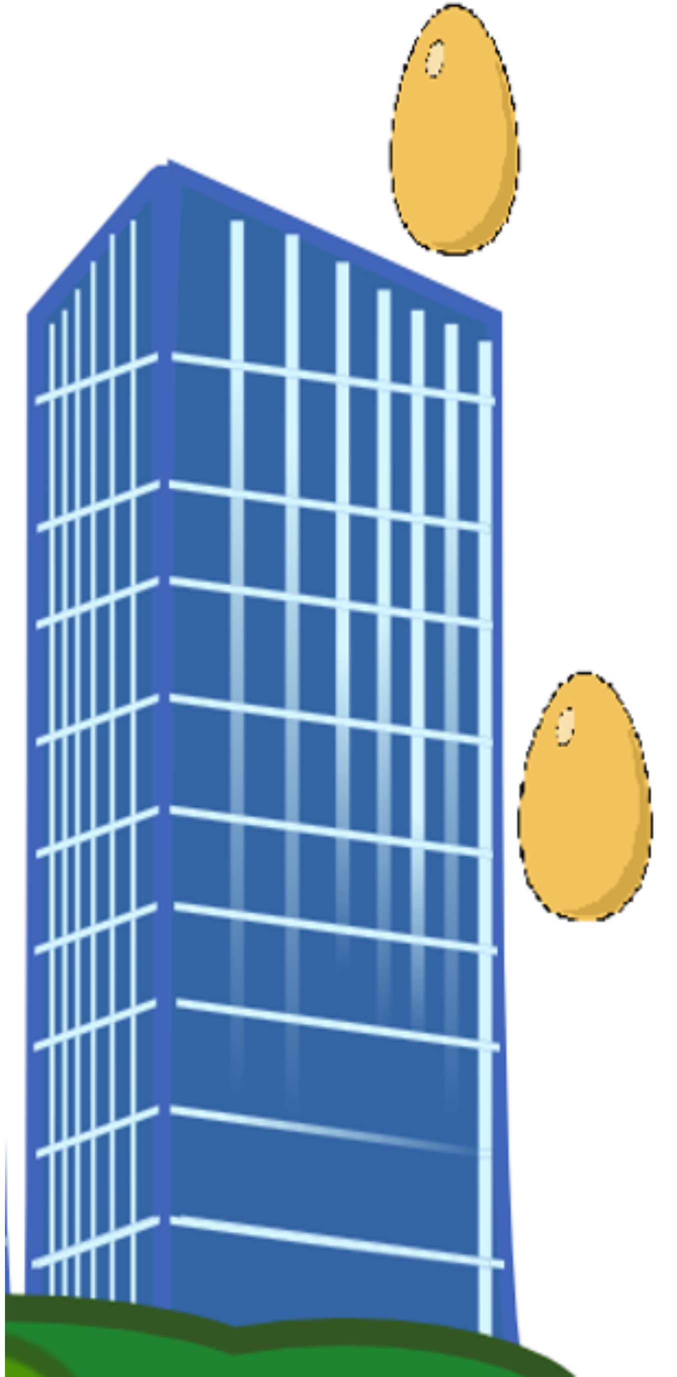
How do you interpret finding n in as few drops as possible?



Interpretation:

How do you interpret finding n in as few drops as possible?

- Minimize worse case.
- Minimize average case.



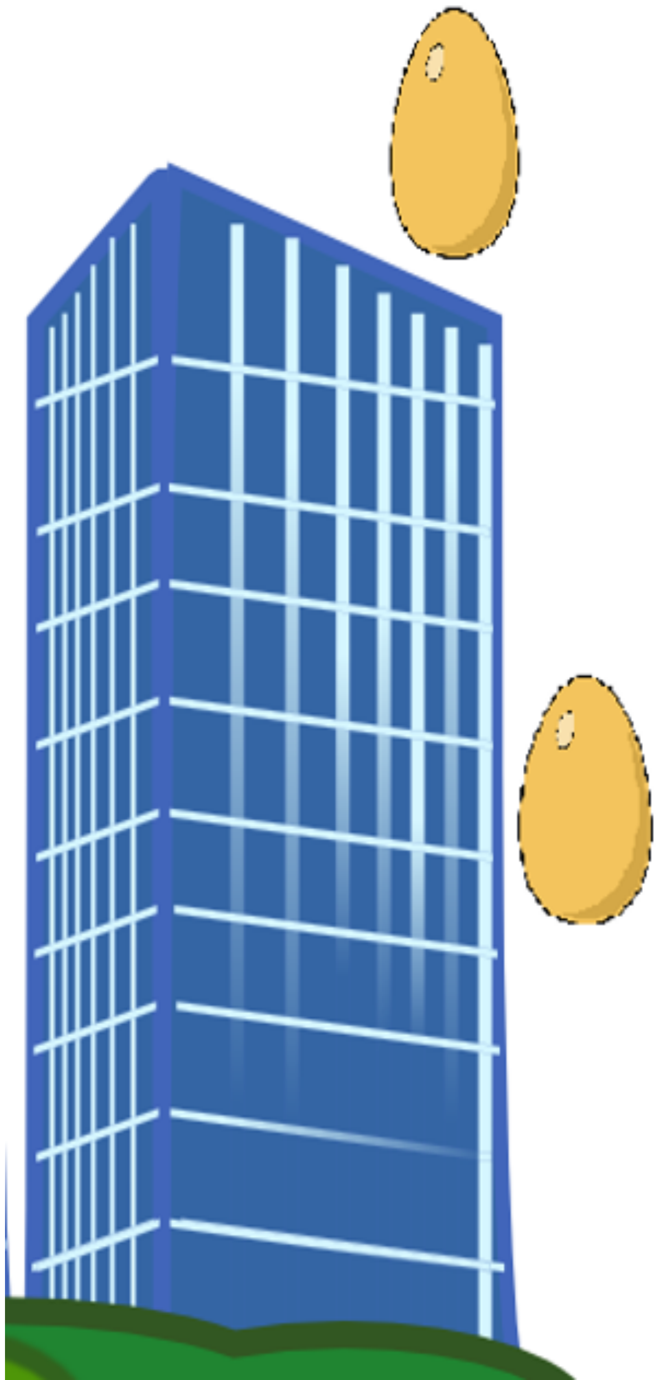
General Advice:

When given a hard problem:

- try to do an easier version first, and
- try to do specific values of parameters.

What is an easier problem?

ONE EGG!

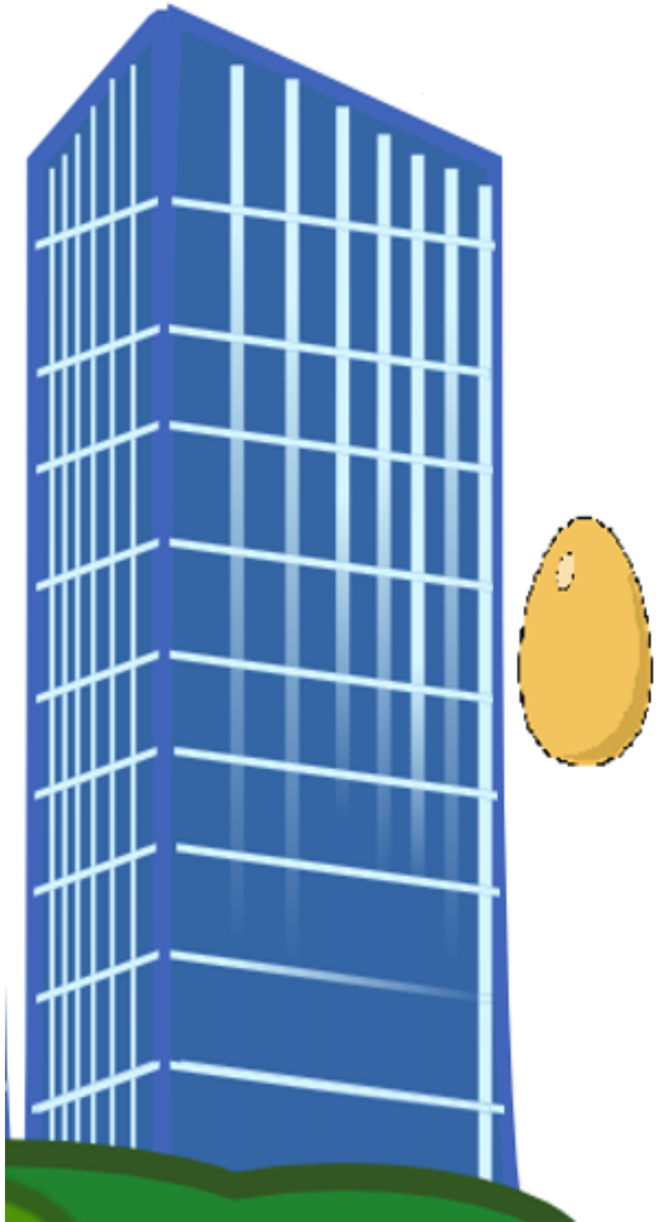


Simple Case: 1 Egg

What is the solution?

have to start with floor 1

Worse case: N drops if have N floors

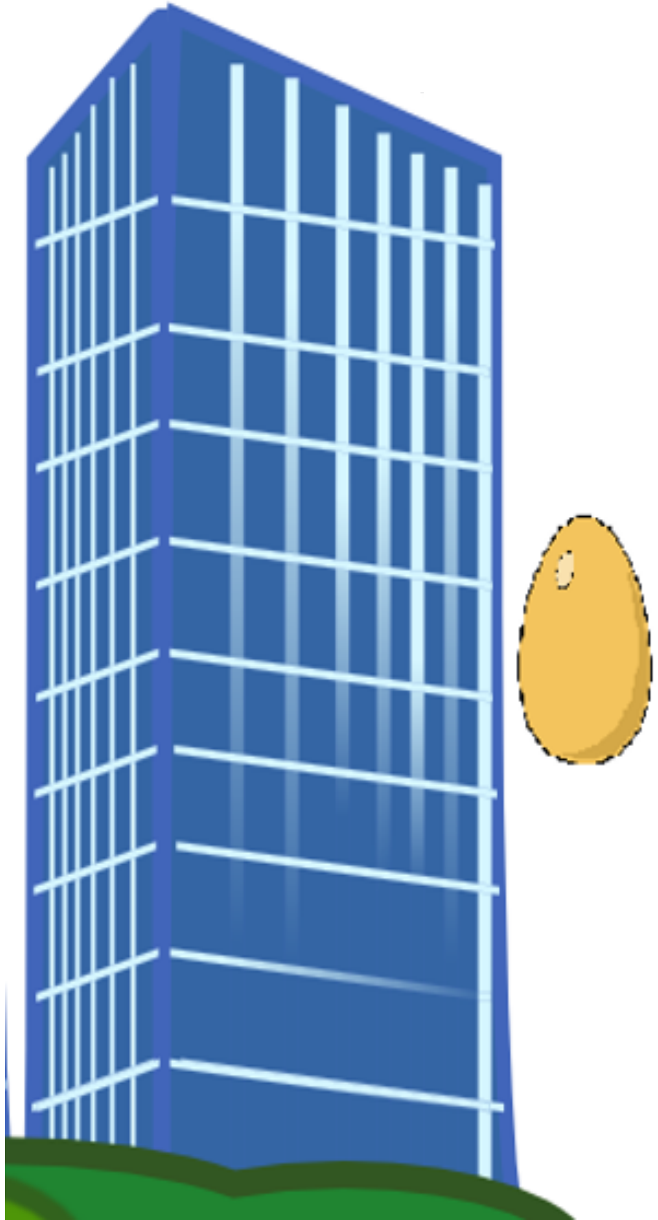


Simple Case: 1 Egg

What is the solution?

Only possibility is go 1, 2, 3, … till break.

Worse case is order N drops.



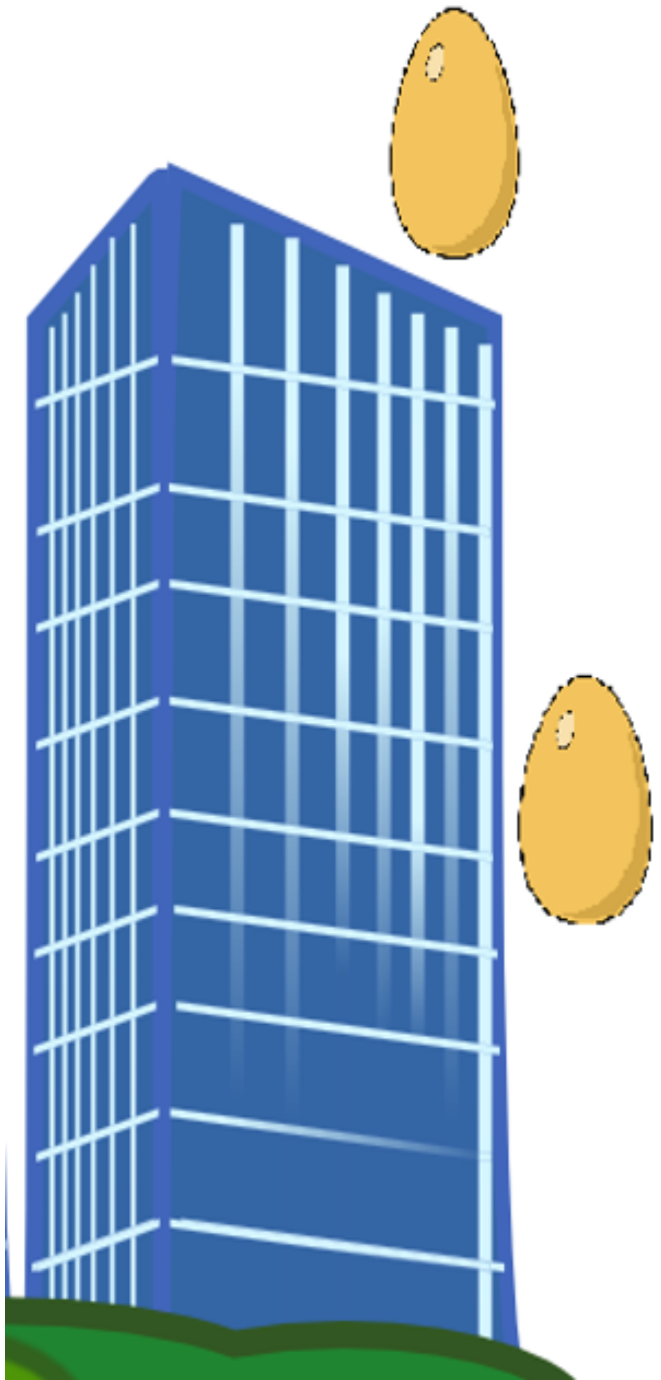
Next Case: 2 Eggs

Once one cracks, reduced to 1 egg case.

What are possible strategies?

Evens: drop 2, 4, 6, 8 till break then go back one

Big Step: Go halfway, if break one at a time else
do half of what is left
If N=100: 50, 75 breaks, 51, 52, 53, ...



Next Case: 2 Eggs

Once one cracks, reduced to 1 egg case.

What are possible strategies?

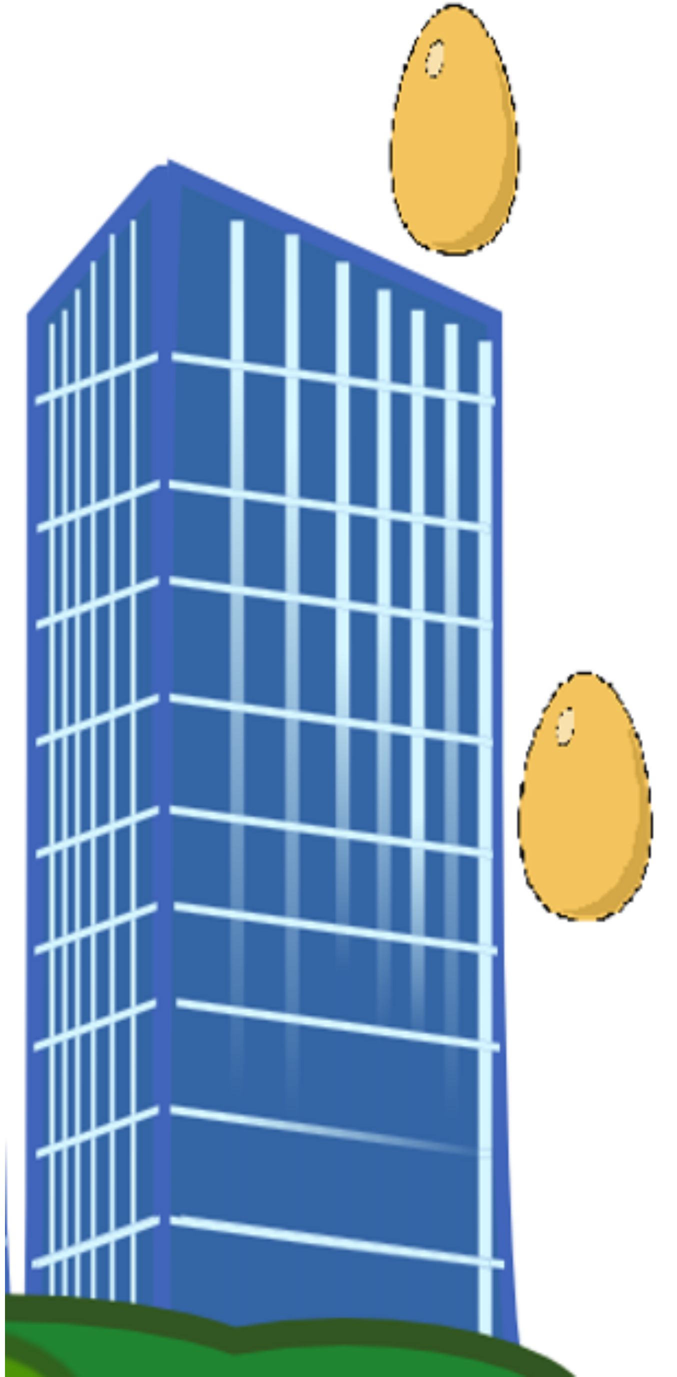
run-time: worse cases with N floors

evens: 2, 4, 6, 8, ..., N ~ $\frac{N}{2}$ drops + 1 more

total is $\frac{N}{2} + 1$ drops

Big Step: breaks immediately (1 drop) then do $\frac{N}{2} - 1$ more

total is $1 + \left(\frac{N}{2} - 1\right) = \frac{N}{2}$



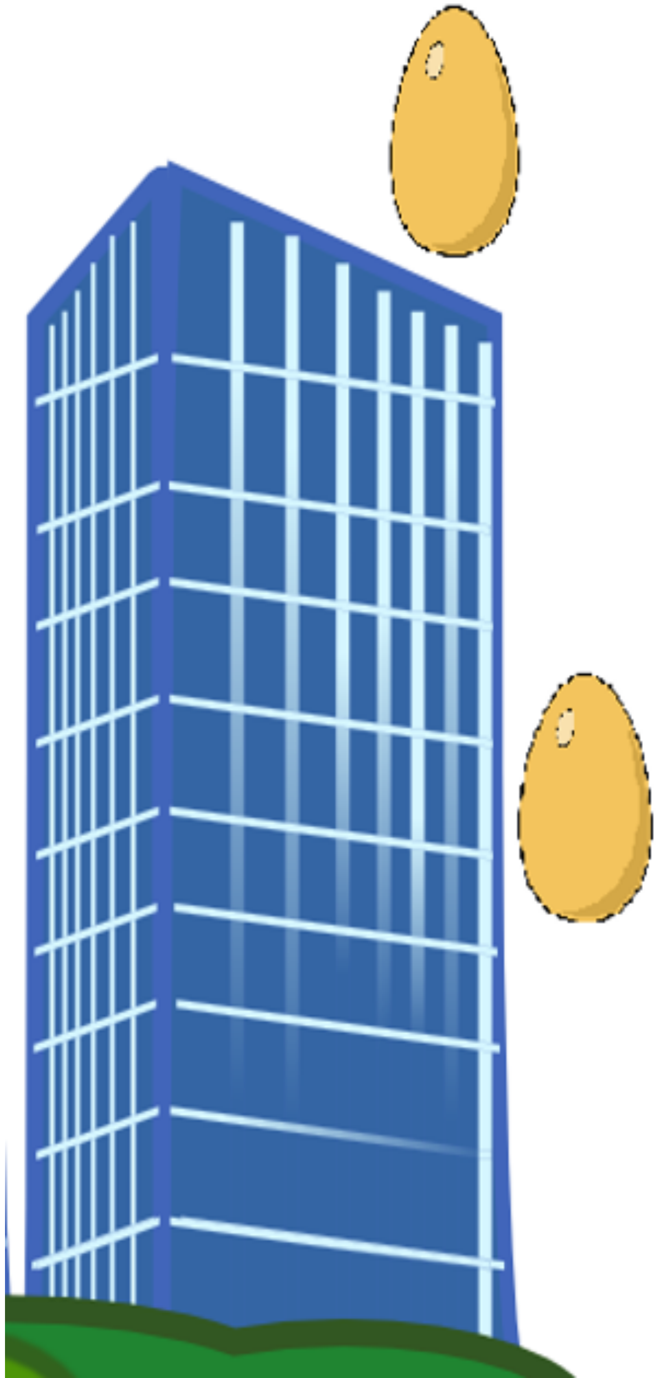
Next Case: 2 Eggs

Once one cracks, reduced to 1 egg case.

What are possible strategies?

Extreme cases:

- Drop every 2nd floor.
- Drop at N/2.
- (more generally drop every x)



Drop every X

Worse case: X, 2X, 3X, 4X, ..., N: # drops $\approx \frac{N}{X}$

now need X-1 drops

so total is $\frac{N}{X} + X - 1$ or $\left(\frac{N}{X} + X\right) - 1$

Minimize $\frac{N}{X} + X$: if X↑ then $\frac{N}{X}$↓ but X↑

if X↓ then $\frac{N}{X}$↑ but X↓

Choose X so that $\frac{N}{X} = X \implies N = X^2$ so $X = \sqrt{N}$

Cost is $\frac{N}{\sqrt{N}} + \sqrt{N} - 1 = \sqrt{N} + \sqrt{N} - 1 \approx 2\sqrt{N}$

Competing Influences

Drop every 2nd floor.

- Once first breaks fast, but could take many drops.
- #Drops = $N/2 + 1$

Drop at N/2

- If doesn't crack eliminate a lot, when crack lot to check.
- #Drops = $1 + (N/2 – 1)$.

Both basically on the order of N/2 drops....

Competing Influences: Balance

Drop every x floors.

Competing Influences: Balance

Reduced to choosing x to minimize

$$\frac{N}{x} + x \,.$$

Competing Influences: Balance

Reduced to choosing x to minimize

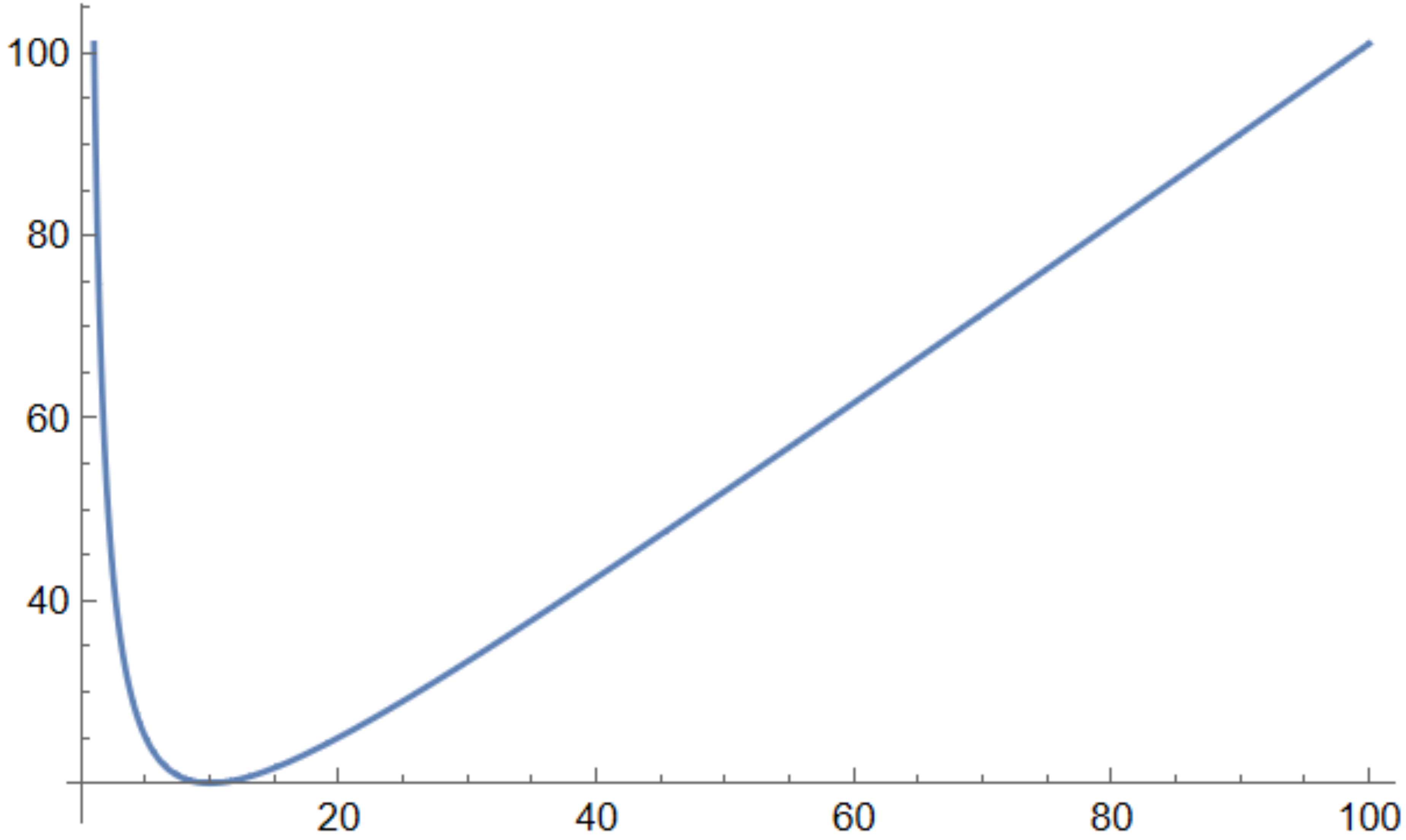
$$\frac{N}{x} + x\,.$$

Set two terms equal to each other to balance:

$$\frac{N}{x} = x \quad so \quad N = x2 \quad or \quad x = N^{1/2}\,.$$

Gives #Drops = $\dfrac{N}{N^{1/2}} + N^{1/2}$ - 1 or about 2 $N^{1/2}$.

```
Plot[100 / x + x, {x, 1, 100}]
```



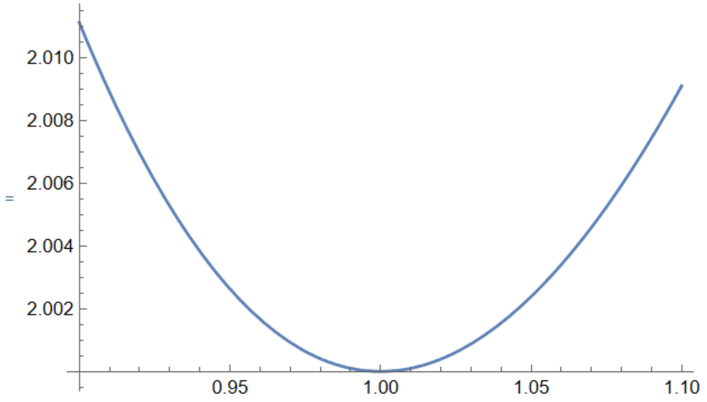
Write x = t $N^{1/2}$ in #Drops = $\frac{N}{x} + x - 1$.

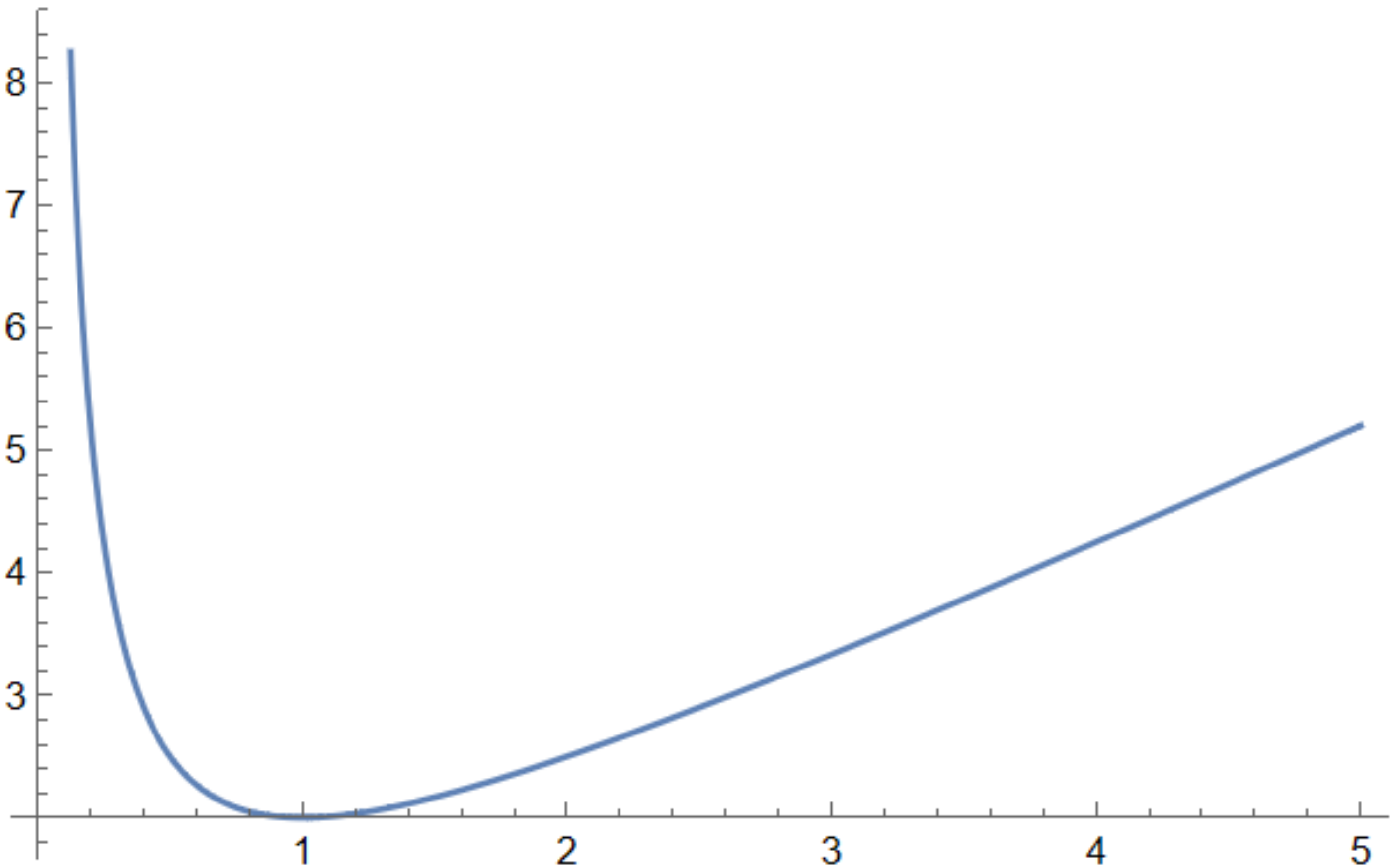
Gives #Drops = $\frac{N}{t\,N^{1/2}} + t\,N^{1/2} - 1$.

```
Plot[1/t + t, {t, 0, 5}]
```

This is just $N^{1/2}\,(\frac{1}{t} + t)$,

so on the order of $N^{1/2}$!





If know calculus: want to minimize $f(x) = N/x + x$:

- Endpoints: $f(1)$ and $f(N)$ are of order N.
- $f'(x) = -N/x^2 + 1$, critical point $f'(x) = 0$ or $x = N^{1/2}$.
- Easily see minimum, or note $f''(x) = 2N/x^3 > 0$.

Balancing Application

Imagine have two algorithms:

- One always takes 1000 seconds.
- One takes 1 second except one in a million inputs take 1,000,000,000 seconds.

if takes over 1,000,000 seconds you die and you get $ 1,000,000,000 if it takes at most 2 seconds

Both take on average approximately 1000 seconds….

do Algorithm 2, if takes > 1 second stop and switch to 1.

Balancing Application

Imagine have two algorithms:

- One always takes 1000 seconds.
- One takes 1 second except one in a million inputs take 1,000,000,000 seconds.

Both take on average approximately 1000 seconds….

…but what if run algorithm 1 and if takes more than 2 seconds on an input switch to first? Average of about 1 second!

Improving Strategy with 2 Eggs

Consider triangular numbers and dynamic rescaline.

- Do not move in constant steps of x floors.
- Do x, then x-1 if doesn't crack, then x-2….
    - Advantage is always same number of drops!
    - Basically if doesn't crack doing 2 egg problem but now with N-x floors (after first drop).

Improving Strategy with 2 Eggs
Consider triangular numbers and dynamic rescaline.

- Do not move in constant steps of x floors.
- Do x, then x-1 if doesn't crack, then x-2….
  - Advantage is always same number of drops!
  - Basically if doesn't crack doing 2 egg problem but now with N-x floors (after first drop).

Example: N = 105 = 14 + 13 + 12 + … + 1:

(1 + 13) or (2 + 12) or (3 + 11) ….

All are 14 drops, better than $2 * 105^{1/2}$ (about 20).

What if we have 3 Eggs? Or k eggs?

drop every x till break, then 2 egg problem:

worse case: do $x, 2x, 3x, \ldots, N$: # drops is $\frac{N}{x}$

now have $x-1$ floors and 2 eggs: drop at $\sqrt{x-1}$

and cost is $2\sqrt{x-1} \simeq 2\sqrt{x}$

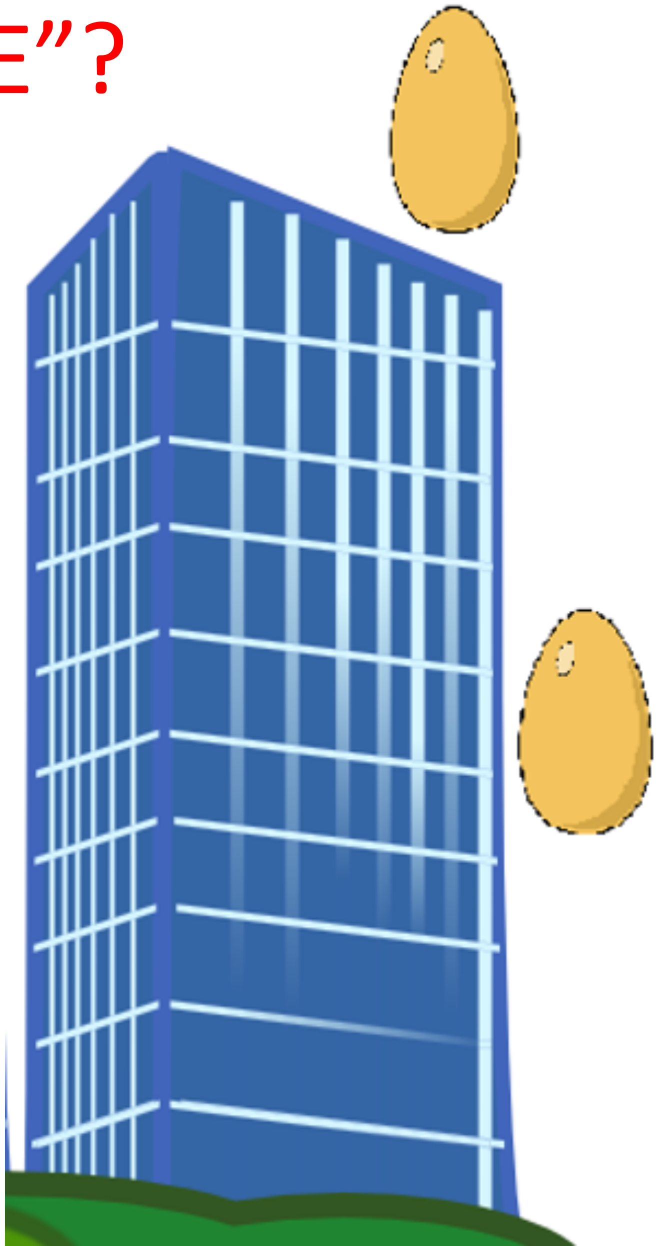
Total Cost is $\frac{N}{x} + 2x^{1/2}$ minimize

if $\frac{N}{x} = 2x^{1/2}$ then $N = 2x^{3/2}$ so $x^{3/2} = \frac{N}{2}$ so $x = \frac{N^{2/3}}{2^{2/3}}$

Cost is $2\frac{N}{x} = 2\frac{N}{N^{2/3}/2^{2/3}} = 2 \cdot 2^{\frac{2}{3}} N^{\frac{1}{3}} = \text{Const} \cdot N^{\frac{1}{3}}$

NEW RESEARCH QUESTION: Email sjm1@williams.edu

What if "TWO-DIMENSIONAL"? Or "THREE"?



- Consider box from $(0,0)$ to (M,N), find special point (m,n) such that if drop at (a,b) with $a < m$ and $b < n$ no damage, otherwise breaks.

- What if breaks only when $m+n > V$?

- What if breaks only when $am + bn > V$?

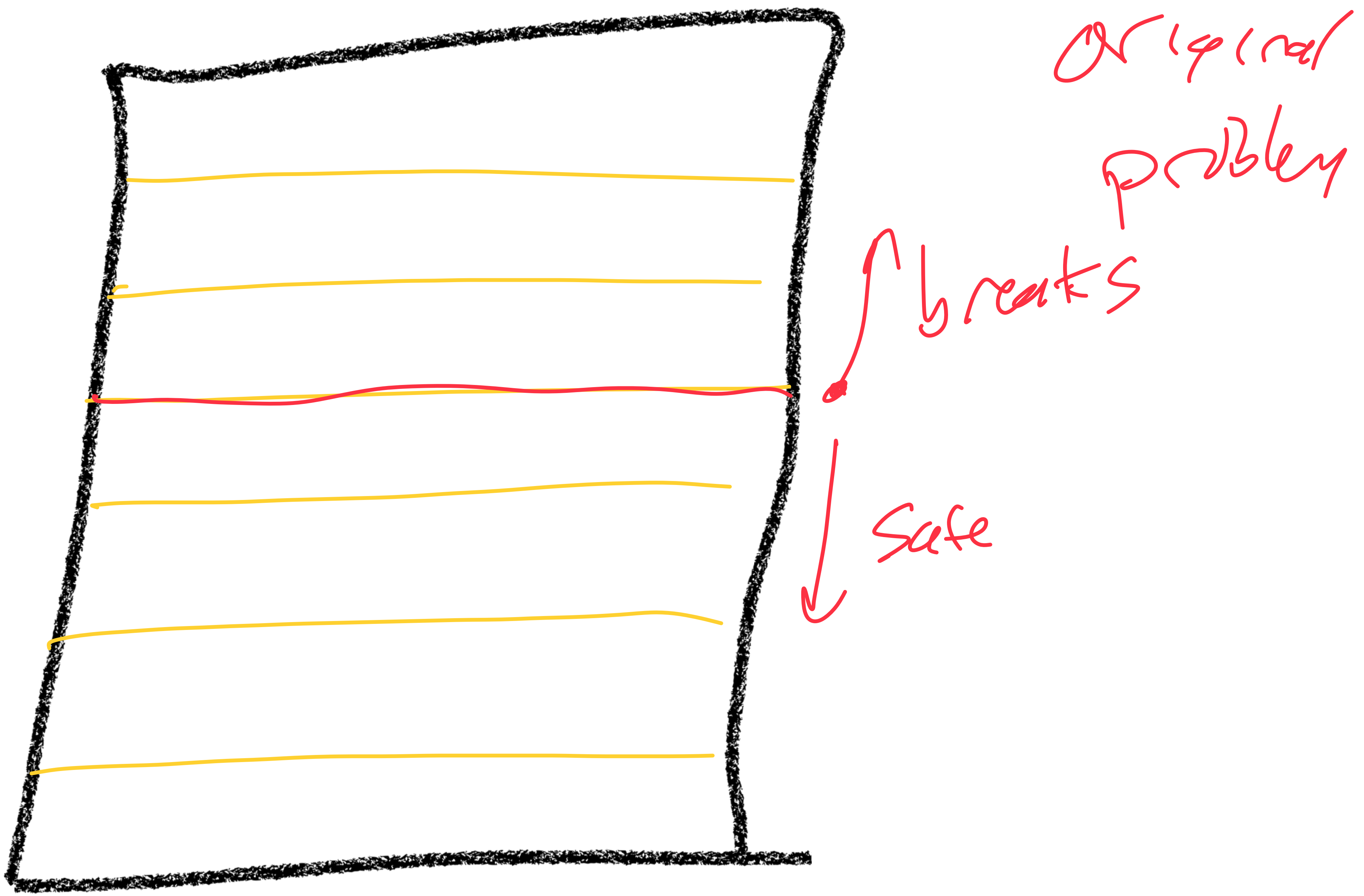
What if we have 3 Eggs? Or k eggs?

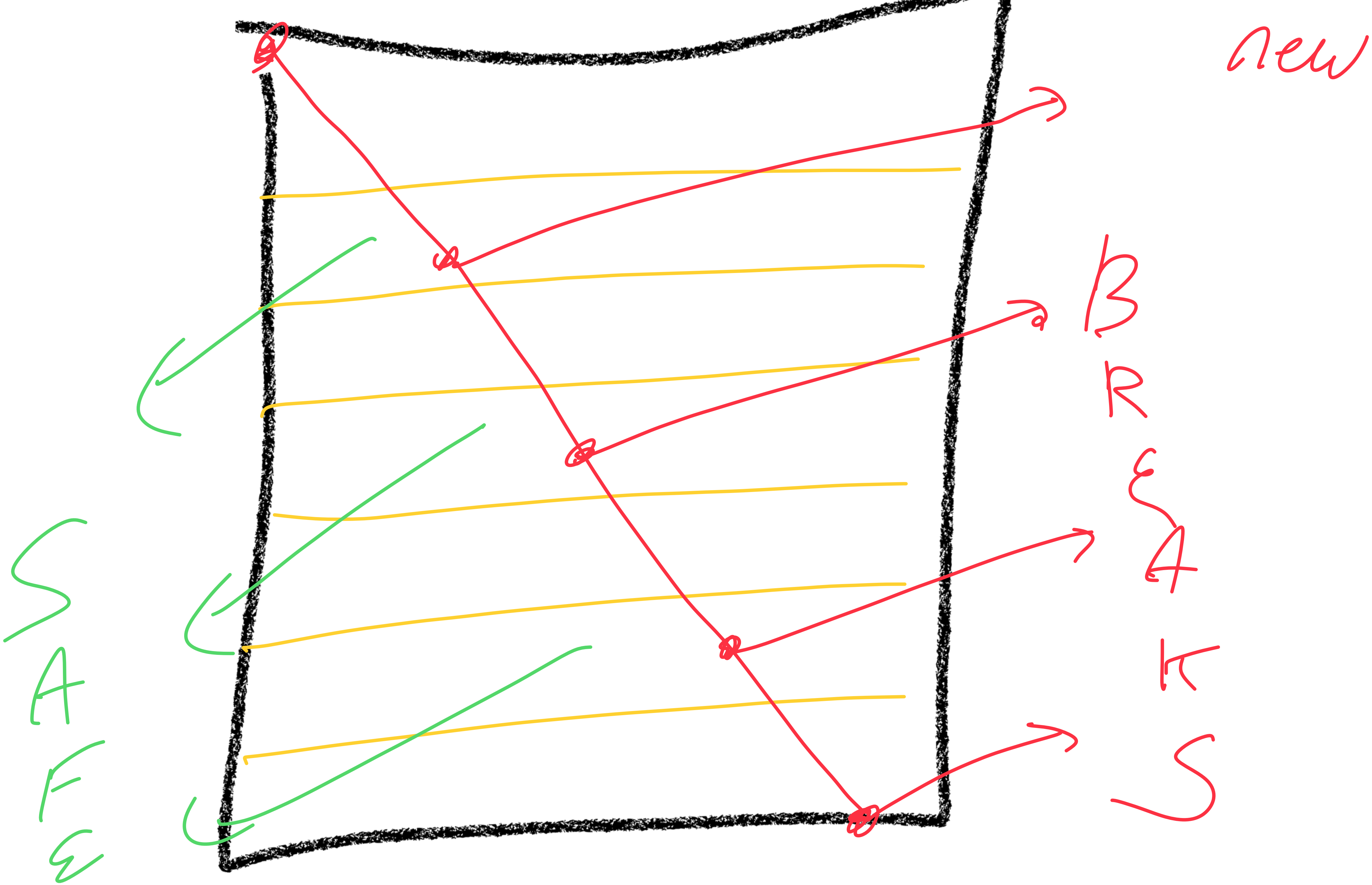
For 3 eggs: once one cracks, 2 egg problem.
If do every x it would be, worse case:

What do we know?

# eggs	drop at	worse case cost
1	$1 = N^0$	$N^{1/1}$
2	$N^{1/2}$	$2N^{1/2}$
3	Const $N^{2/3}$	Const $N^{1/3}$
k	Cost $N^{\frac{k-1}{k}}$	Const $N^{1/k}$

CONJECTURE


original
problem
breaks
safe


new
BREAKS
SAFE

BIG TAKEAWAYS

- what to study: worse case or average case?
- combining algorithms
- balance issues
- Start simple, build intuition
- reduce to easier cases