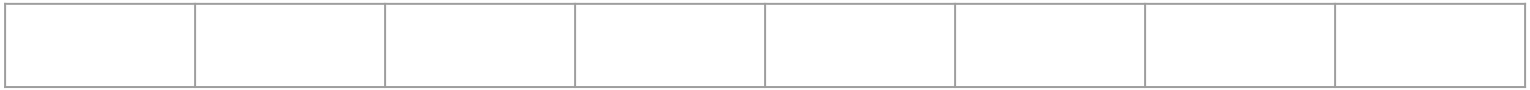
PSO 13

Amortization and Randomization

Amortization: doubling array

Supporting insertion into static arrays in O(1) *amortized* time

Alloc(n)



Amortization: doubling array

Supporting insertion into static arrays in O(1) *amortized* time

Alloc(n)
Insert(1)

Amortization: doubling array

Supporting insertion into static arrays in O(1) *amortized* time

Alloc(n)
Insert(1)
Insert(2) $O(1)$

Amortization: doubling array

Supporting insertion into static arrays in O(1) *amortized* time

Alloc(n)
Insert(1)
Insert(2)
…
Insert(n - 1)

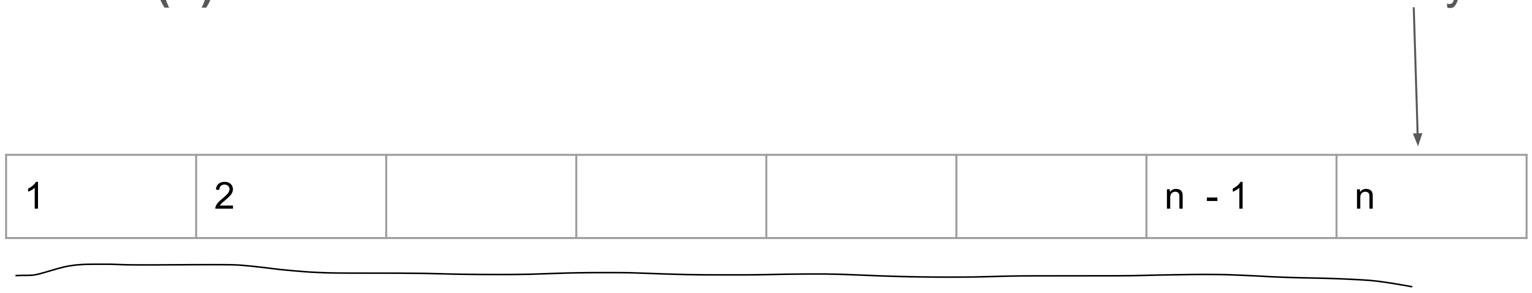
All O(1) worst case operations so far…

Amortization: doubling array

Supporting insertion into static arrays in O(1) *amortized* time

Alloc(n)
Insert(1)
Insert(2)
…
Insert(n - 1)
Insert(n)

If Array.size == Array.capacity:
    alloc(2n)
    //move all elts to new array

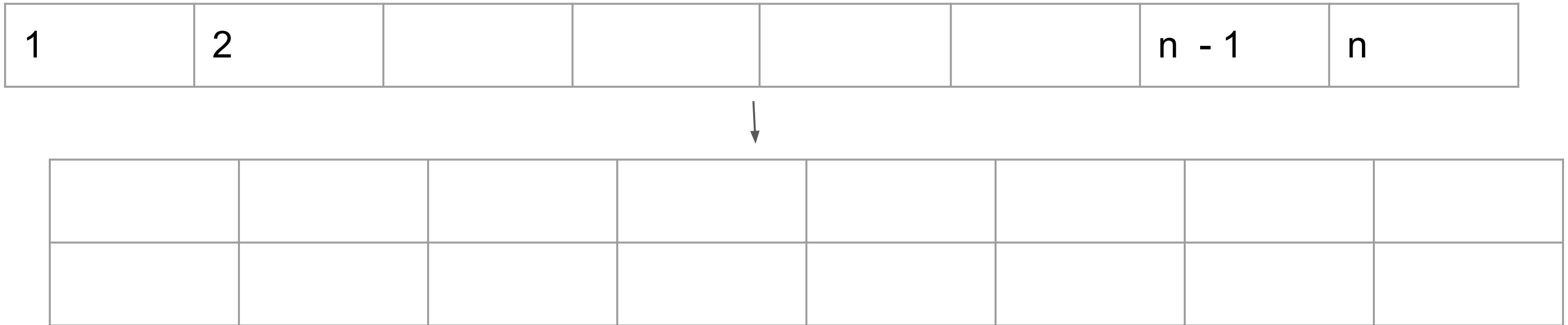


Amortization: doubling array

Supporting insertion into static arrays in O(1) *amortized* time

Alloc(n)
Insert(1)
Insert(2)
…
Insert(n - 1)
Insert(n)

If Array.size == Array.capacity:
    alloc(2n)
    //move all elts to new array



Amortization: doubling array

Supporting insertion into static arrays in O(1) *amortized* time

O(n) worst case, but supposedly constant amortized.. why?

Alloc(n)
Insert(1)
Insert(2)
…
Insert(n - 1)
Insert(n)

```
If Array.size == Array.capacity:
      alloc(2n)
      //move all elts to new array
```

1	2	…				n - 1	n

↓

1	2	…				n - 1	n

Amortization with Coins

Banker's method: every elt gets some # coins, used to pay for expensive operations

Alloc(n)

RULES
Initially 0 coins in the system
coins never drops below 0

If this is true, you can conclude..

coins per elt = k ⇒ O(k) amortized cost
$O(1)$

Amortization with Coins

Banker's method: every elt gets some # coins, used to pay for expensive operations

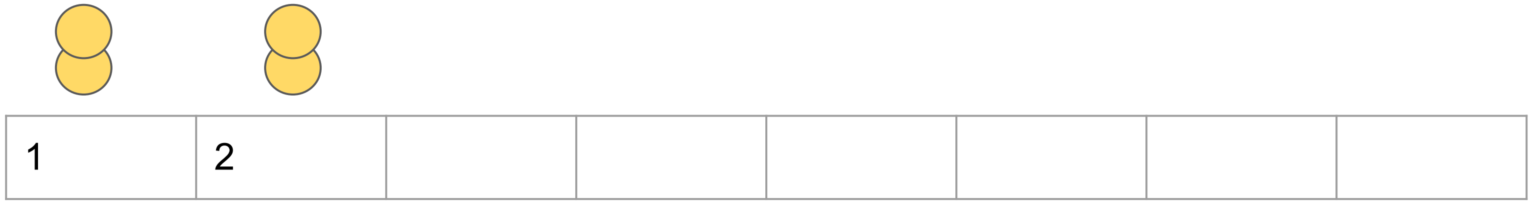
Alloc(n)
Insert(1)

(Allocate C coins, we will find specific C later..)

Amortization with Coins

Banker's method: every elt gets some # coins, used to pay for expensive operations

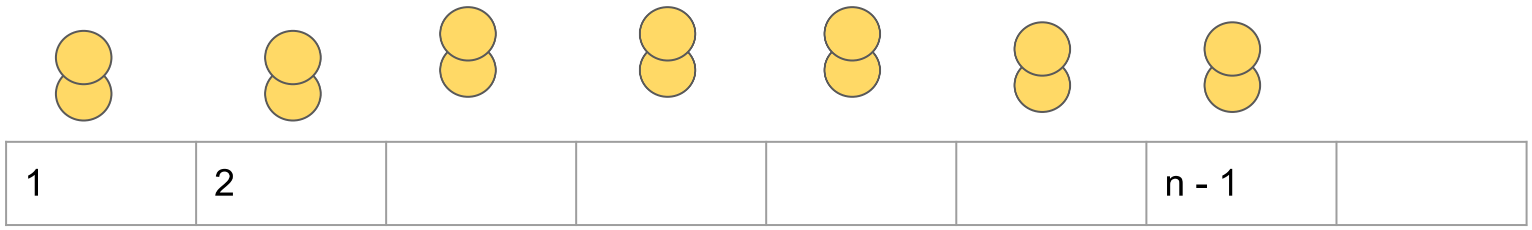
Alloc(n)
Insert(1)
Insert(2)



Amortization with Coins

Banker's method: every elt gets some # coins, used to pay for expensive operations

Alloc(n)
Insert(1)
Insert(2)
…
Insert(n - 1)

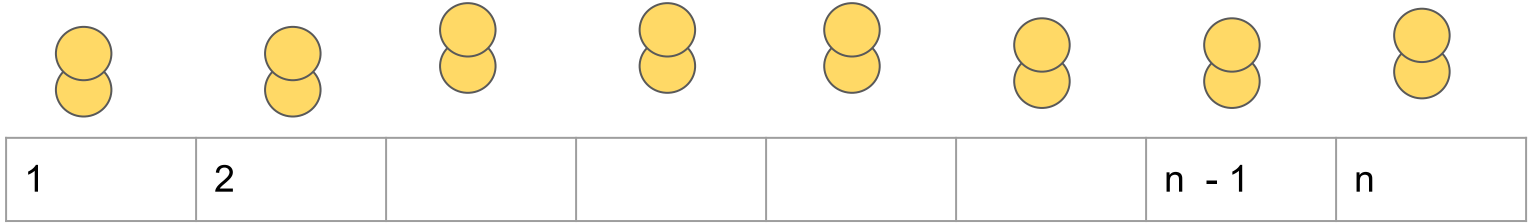


Amortization with Coins

Banker's method: every elt gets some # coins, used to pay for expensive operations

Alloc(n)
Insert(1)
Insert(2)
…
Insert(n - 1)
Insert(n)

```
If Array.size == Array.capacity:
        alloc(2n)
        //move all elts to new array
```

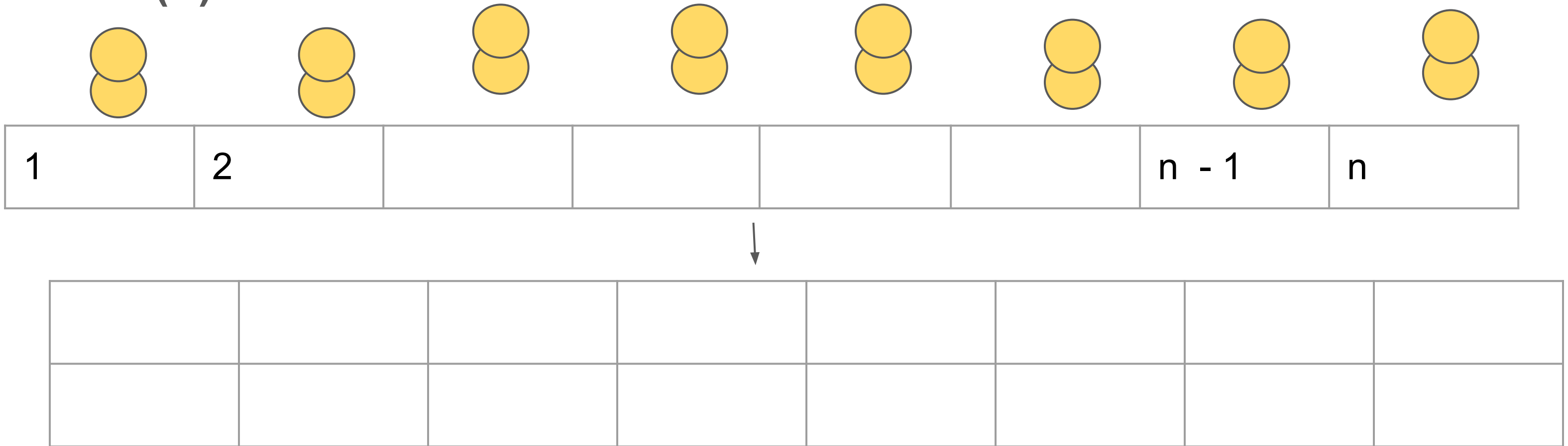


Amortization with Coins

Banker's method: every elt gets some # coins, used to pay for expensive operations

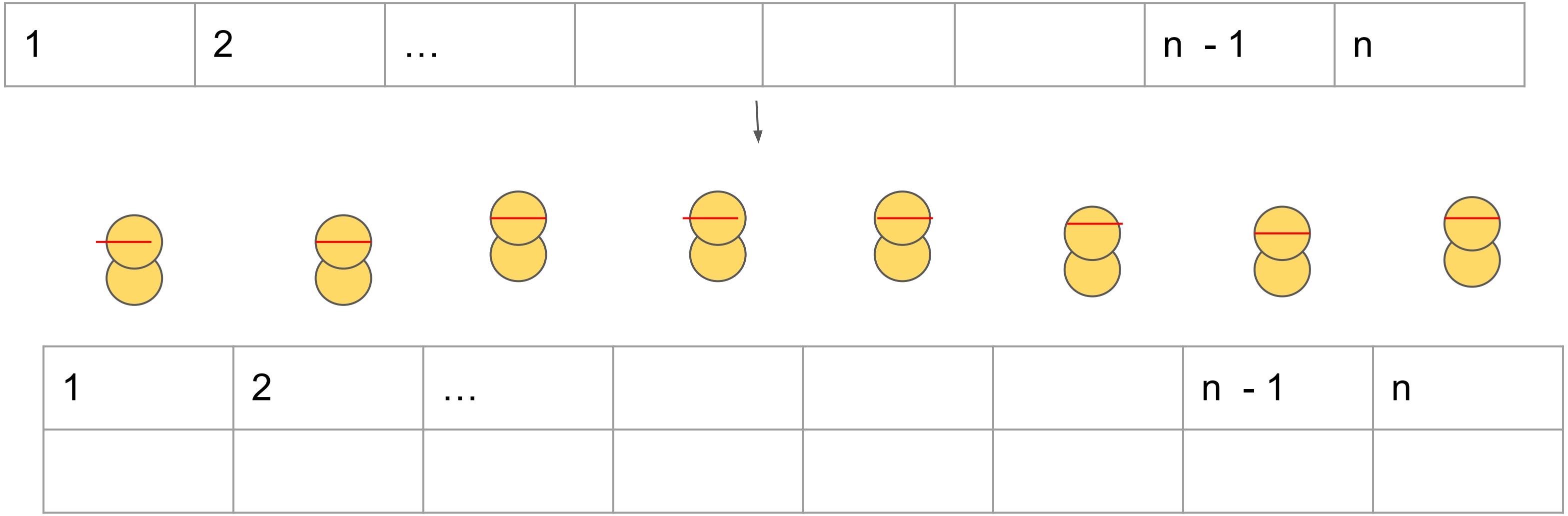
Alloc(n)
Insert(1)
Insert(2)
...
Insert(n - 1)
Insert(n)

If Array.size == Array.capacity:
alloc(2n)
//move all elts to new array



Amortization with Coins

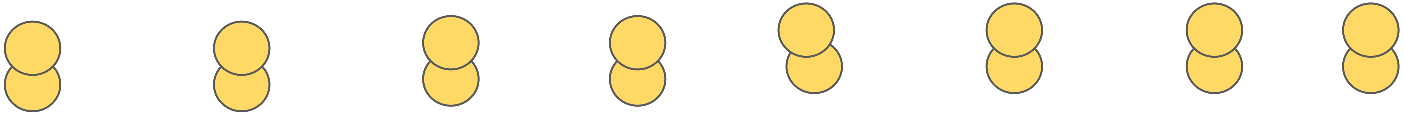
We only need one coin per elt to pay for the doubling! (But is 1 enough?)



Double -> Double (Expensive op after expensive op)

Add n more items

1	2	…				n - 1	n
(n + 1)	(n+2)						2n



Double -> Double (Expensive op after expensive op)

Double (need to pay 2n)

1	2	…				n - 1	n
(n + 1)	(n+2)						2n

1	2	…				n - 1	n
(n + 1)	(n+2)						2n

Double -> Double (Expensive op after expensive op)

2 coins per item

Can use solely the “new” coins

1	2	…				n - 1	n
(n + 1)	(n+2)						2n

1	2	…				n - 1	n
(n + 1)	(n+2)						2n

How do you prove this?

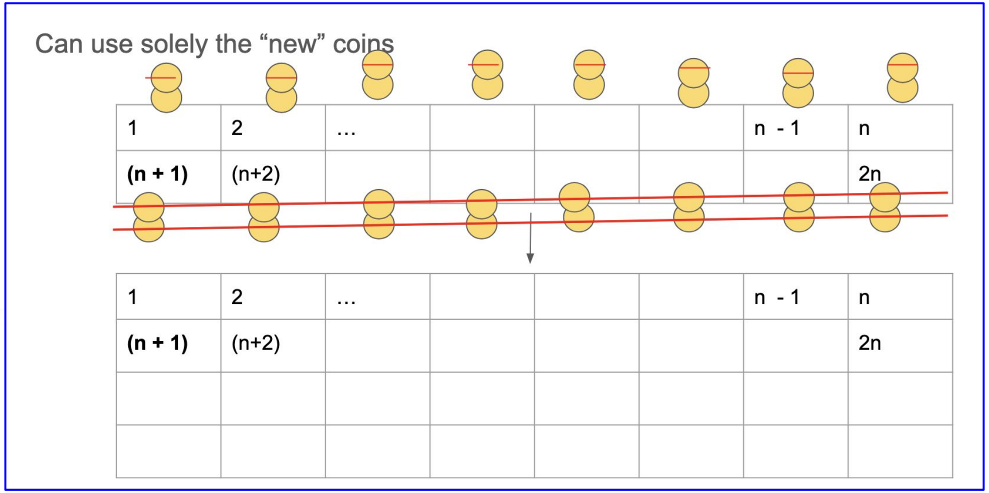
First, the rules are followed.

Then,

C =

• Allocate C coins per inserted element

• Inserts w/o resizing – O(1) time



RULES

ally 0 coins in the system

oins never drops below 0

is true, you can conclude..

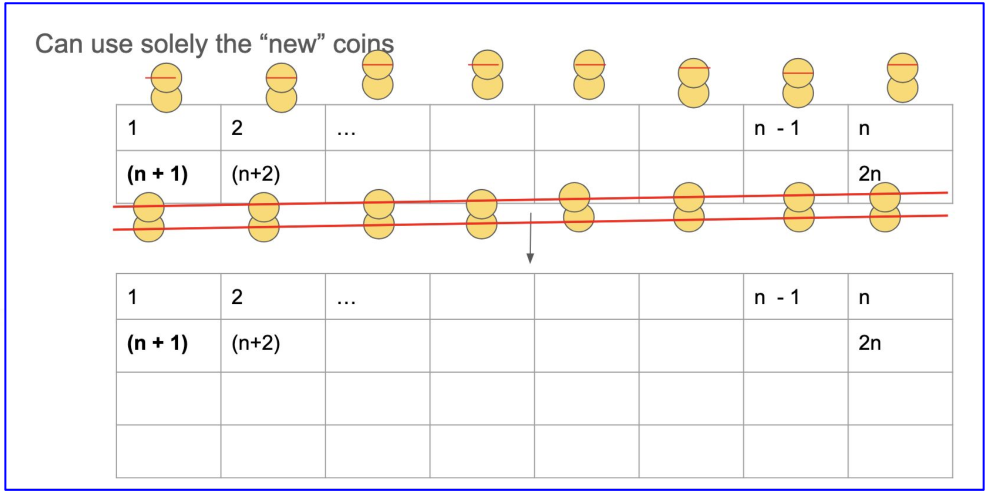
How do you prove this?

First, the rules are followed.

Then,

$C = 2$

- Allocate C coins per inserted element
- Inserts w/o resizing – $O(1)$ time
- Inserts w/ resizing:
  - SPS we have n elts currently 



RULES

ally 0 coins in the system

oins never drops below 0

is true, you can conclude..

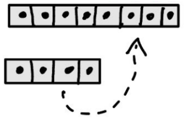
How do you prove this?

First, the rules are followed.

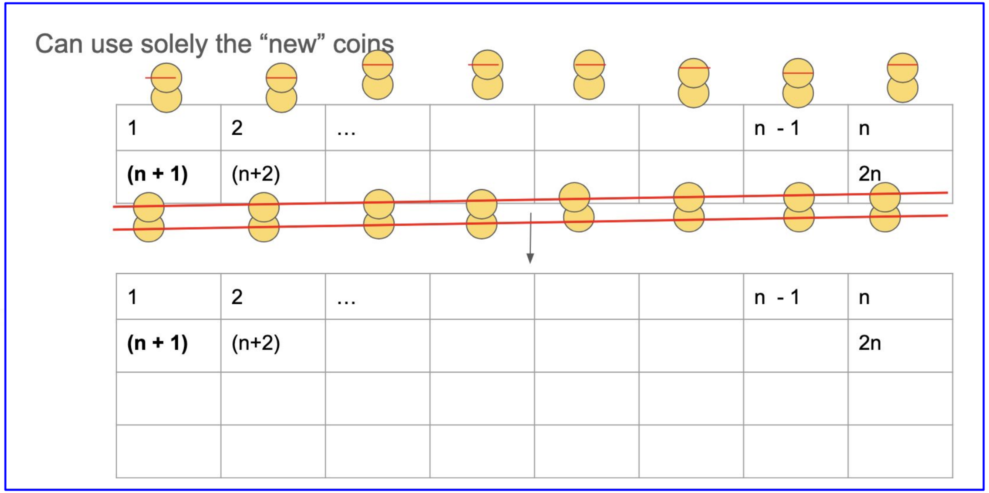
Then,

$C = 2$

- Allocate C coins per inserted element
- Inserts w/o resizing – $O(1)$ time
- Inserts w/ resizing:
  - SPS we have n elts currently
  - We have successfully resized when we had $n/2$ elts



(induction).



RULES

ally 0 coins in the system

oins never drops below 0

is true, you can conclude..

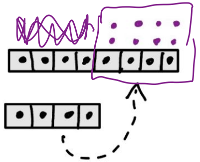
How do you prove this?

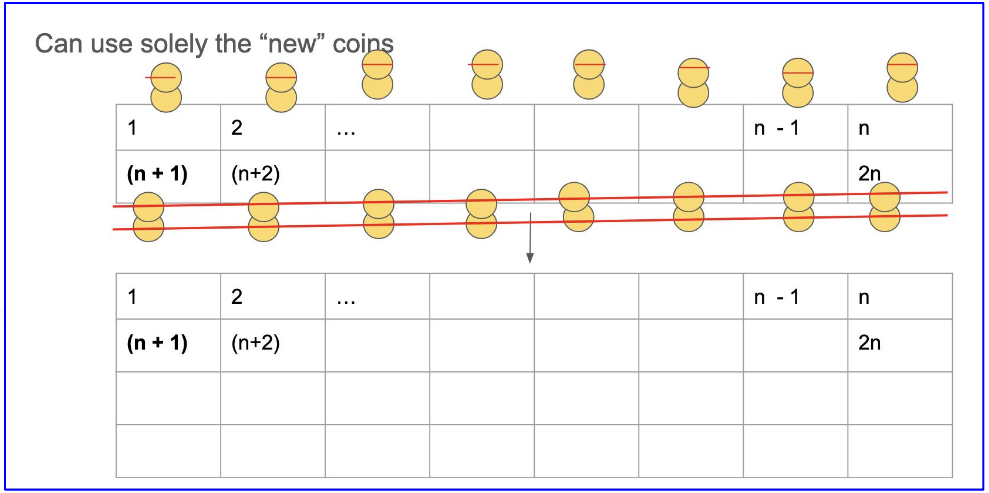
First, the rules are followed.

Then,

$C = 2$

- Allocate C coins per inserted element
- Inserts w/o resizing – $O(1)$ time
- Inserts w/ resizing:
  - SPS we have n elts currently
  - We have successfully resized when we had n/2 elts
  - There are n/2 elts leftover with C unused coins





RULES

ally 0 coins in the system

oins never drops below 0

is true, you can conclude..

How do you prove this?

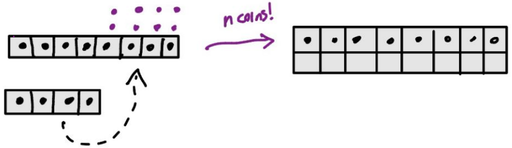
First, the rules are followed.

Then,

C = 2

- Allocate C coins per inserted element
- Inserts w/o resizing — O(1) time
- Inserts w/ resizing:
  - SPS we have n elts currently
  - We have successfully resized when we had n/2 elts
  - There are n/2 elts leftover with C unused coins
  - C × n/2 = n coins to use for copying

(POG)

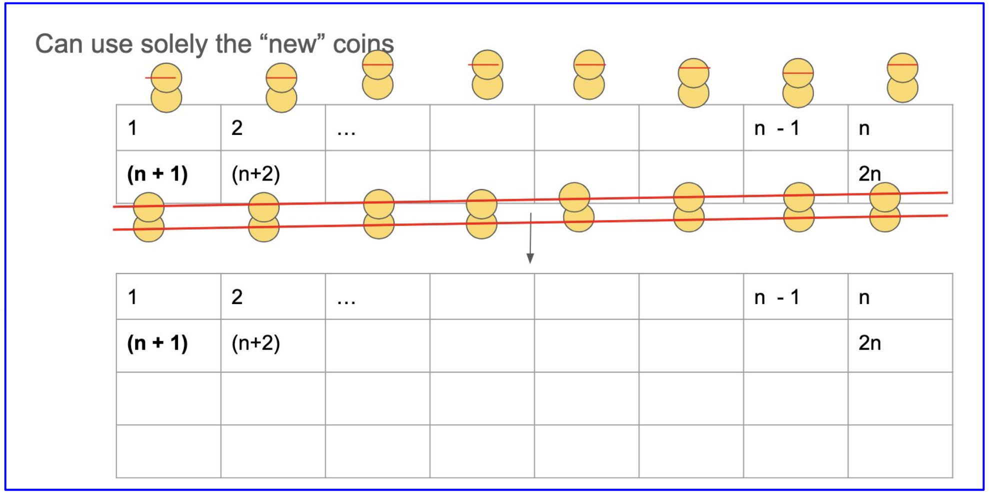


need n coins

have (n/2) elements w/ C coins

$C(n/2) \geq n$

C = 2



RULES

ally 0 coins in the system

oins never drops below 0

is true, you can conclude..

(Re)Introduction to Probability

The study of random variables, events, and expectation

(Re)Introduction to Probability

The study of random variables, events, and expectation

Coin flip (H/T)

(Re)Introduction to Probability

The study of random variables, events, and expectation

Coin flip (H/T)	One flip result is heads/tails	
	With probability	
	$Pr[heads] = 1/2$	
	$Pr[tails] = 1/2$	

(Re)Introduction to Probability

The study of random variables, events, and expectation (**or average)**

Coin flip (H/T)

One flip result is heads/tails

With probability

E[flip result] = .5 heads

|| heads

$(.5)(1)$

$+ (.5)(0)$

$= .5$

More formally…

(Re)Introduction to Probability

The study of random variables, events, and expectation (**or average)**

$X = \begin{cases} 1 \text{ if heads,} \\ 0 \text{ if tails} \end{cases}$	$(X = 1)$ or $(X = 0)$	$E[X] = .5$
	$Pr[X = 1] = .5$ $Pr[X = 0] = .5$	
Random algorithm A With output A(input) = 0 or 1 or 2		

How should you think about probability?

My interpretation: mass and proportion

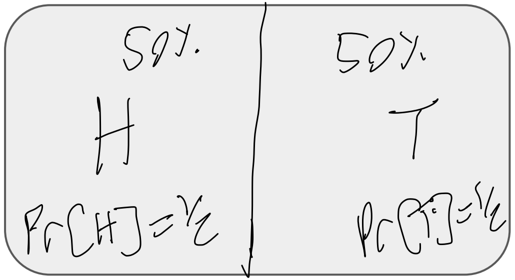
Ex: Let X and Y be the random variables of two (different) coin flips

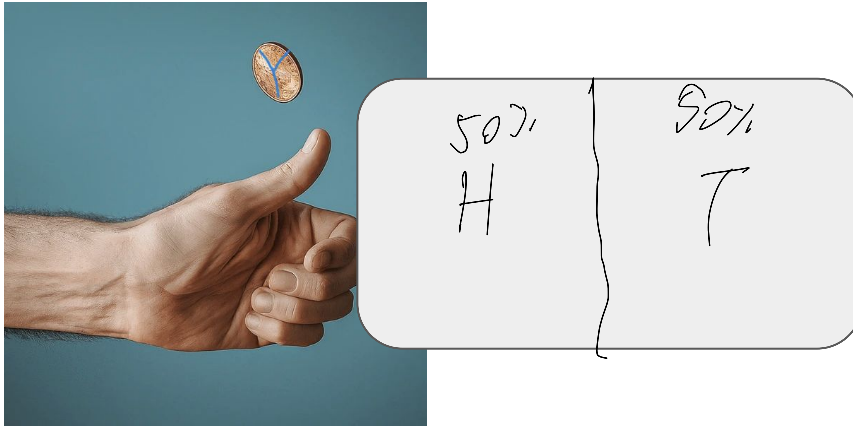
How should you think about probability?

My interpretation: mass and proportion

Ex: Let X and Y be the random variables of two (different) coin flips

Each have their own possibilities (**probability space**)





How should you think about probability?

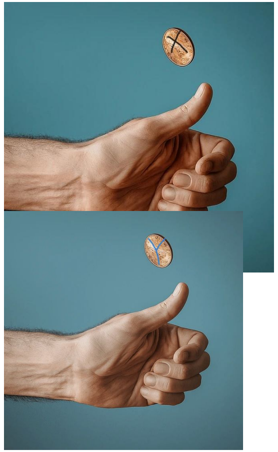
My interpretation: mass and proportion

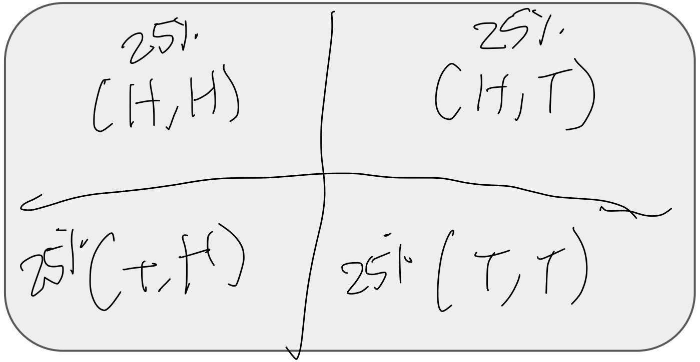
Ex: Let X and Y be the random variables of two (different) coin flips

Each have their own possibilities (**probability space**)

What is the probability space of RV Z = (X,Y)? =

(H, H)
(H, T)
(T, H)
(T, T)





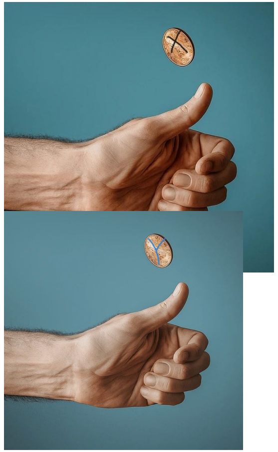
How should you think about probability?

My interpretation: mass and proportion

Ex: Let X and Y be the random variables of two (different) coin flips

Each have their own possibilities (**probability space**)

What is the probability space of RV Z = (X,Y)?



(H,H)	(H,T)
(T,H)	(T,T)

Pr[X = H, Y = T] = 1/4

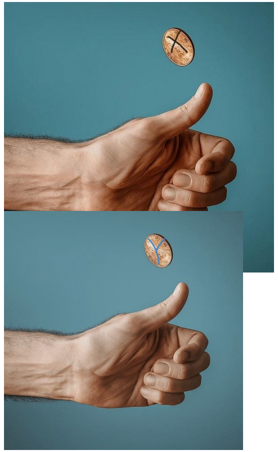
How should you think about probability?

My interpretation: mass and proportion

Ex: Let X and Y be the random variables of two (different) coin flips

Each have their own possibilities (**probability space**)

What is the probability space of RV Z = (X,Y)?



(H,H)	(H,T)
(T,H)	(T,T)

Pr[X = H, Y = T] = 1/4
Pr[X = T, Y = H] = 1/4
Pr[X = H or Y = H] = 3/4.

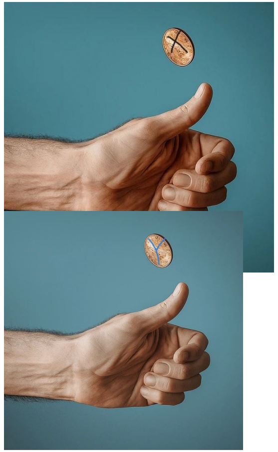
How should you think about probability?

My interpretation: mass and proportion

Ex: Let X and Y be the random variables of two (different) coin flips

Each have their own possibilities (**probability space**)

What is the probability space of RV Z = (X,Y)?



(H,H)	(H,T)
(T,H)	(T,T)

Pr[X = H, Y = T] = 1/4
Pr[X = T, Y = H] = 1/4
Pr[X = H or Y = H] = 3/4

How should you think about probability?

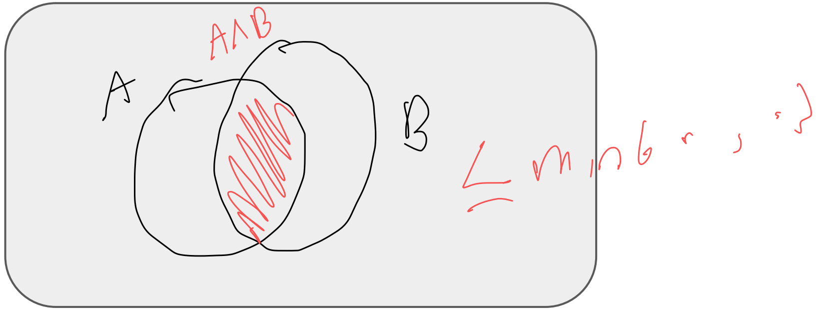
My interpretation: mass and proportion

More generally, two algorithms A,B outputting either 0,1,2 with R.V.s X,Y

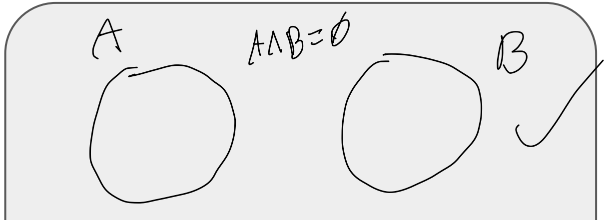
Exercise 25.1. Prove or disprove: For any two events A, B,

$$\mathbf{P}[A \wedge B] \leq \min\{\mathbf{P}[A], \mathbf{P}[B]\}.$$

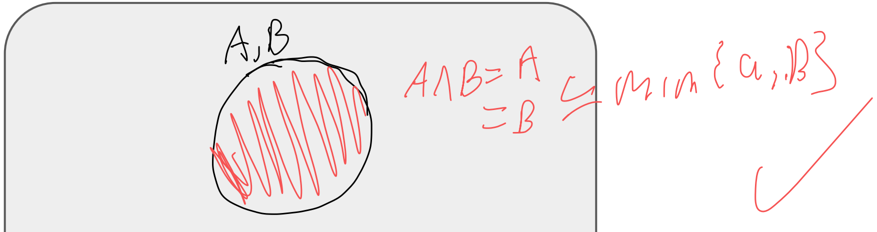
Intuition is incredibly important for probability. What can the prob. space look like?



or



Xtreme case 1



Xtreme case 2