

# USA Tutor

Dishwashing

Analysis by David Yang

# Sample

5

4 5 2 3 1



4

5

2

3

1

USA Tutor



# Step 1

Put the 4 onto a new stack

Why: Only choice



4

5

2

3

1



~

4



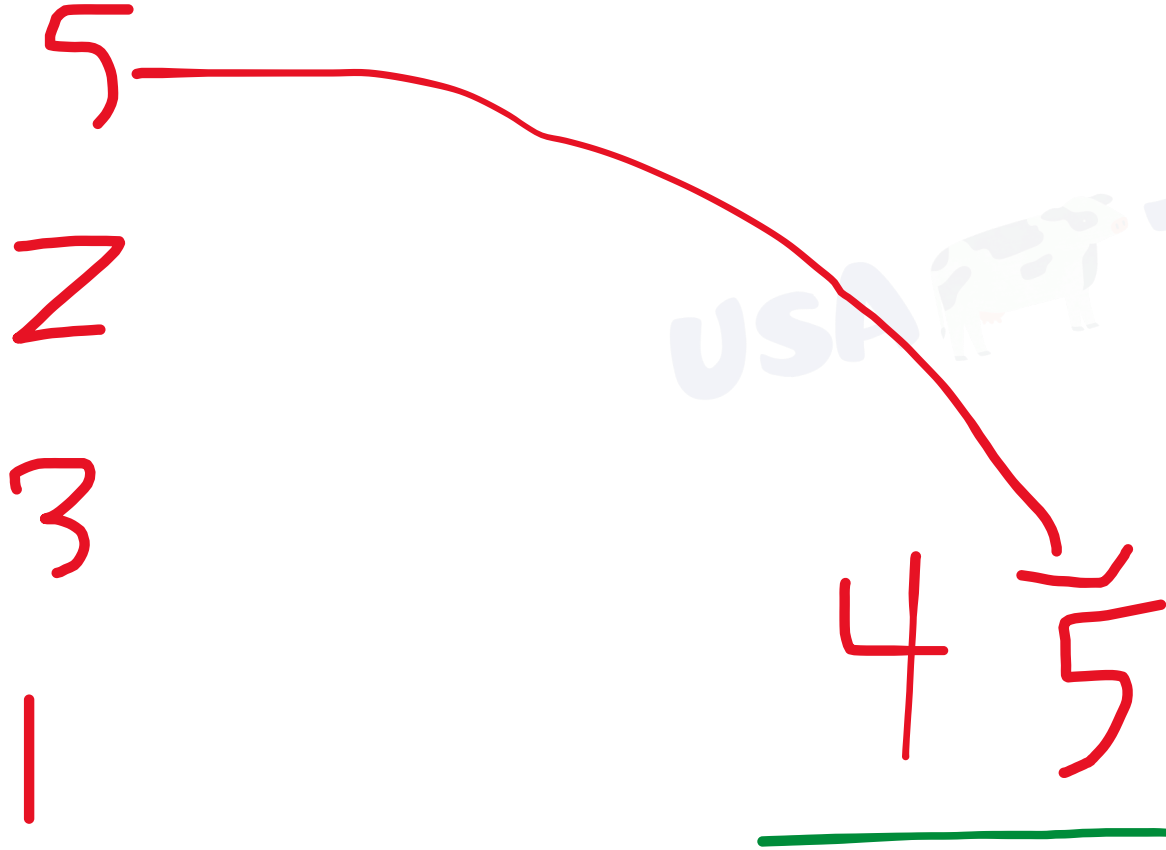
## Step 2

Put 5 onto a new stack

Why: You cannot put a 5 on a 4

Since you keep track of the max of numbers on the Elsie Stack

-> You don't want to add more plates than necessary



# Step 3

Put 2 on top of 4 stack

Why: You can't put it on top of 5 or on a new stack





Z

3

1



2

4

5



# Step 4

Move 2 to Elsie Stack

Why: Make room for the 3



USA Tutor

3

1

2

4

5

USA  Tutor

3

1

2

4

5

2



USA  Tutor

3

1

4 5

---

2

---

# Step 5

Move the 3 on top of the 4 stack

Why: because you can't put it on the 5 or on a new stack

And you don't have to move 4/5 on Elsie, so don't

USA Tutor

3

3

4

5

2

# STOP

2 is on the Elsie Stack

1 is our current number

$1 < 2$ , therefore

we will never have a solution if we decided to take 1

so we quit



USA Tutor

3

4

5

2

STOP

USA  Tutor

3

4

5

---

1

2

# Implementation Details

Casework:

Put current node on stack

Move node to Elsie

Delete nodes



# Put node on stack

What stack should we put it on?

Only 1 will work

How to pick?

You can just iterate/binary search

# Move node to Elsie

Just do `min(curnode, minnode)`

You actually do not have to store Elsie's stack

All that matters is the minnode

# Deleting Nodes

We have to delete starting from the leftmost stack (think about this)

And to delete we do the operations listed in previous slide



# Coding Nuances

1. Determining the stack to place on
2. Storing a pointer to the leftmost stack
3. Dealing with adjacency lists

