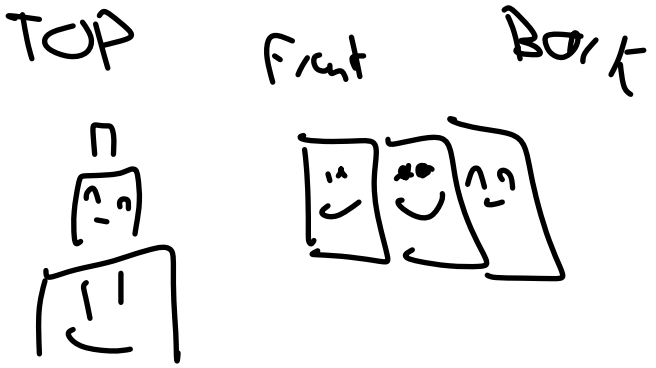
Algorithms and Data Structures for Data Science
lab_quacks

CS 277
Brad Solomon

February 16, 2024

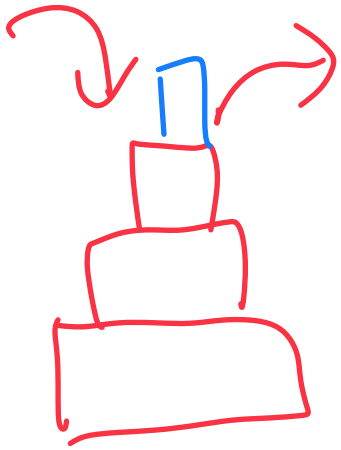


UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN
Department of Computer Science

Learning Objectives

Practice using the stack and the queue

Stack ADT



Order: Last-in, First-Out

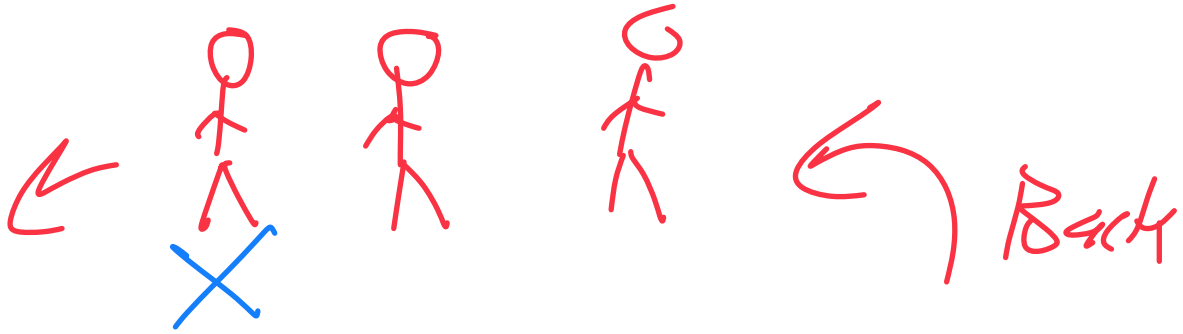
Operations:

Push( ) - Add to top of stack

Pop( ) - Remove and return the top item

Top( ) - Looks at top items value

Queue ADT



Order: First-in, First-Out

Operations:

Enqueue( ) – Adds to back of queue

Dequeue( ) – Removes and returns from front

Front( ) – Gets value at front

Programming Practice

For each problem consider:

sub problems

Input → output

Do I know what the problem is asking me to do?

What values in the stack or queue are relevant? How can I access them?

↳ only have top & front

Do I need any additional data structures to solve the problem?

isBalanced(queue)

1) Same amount of "[" "]"

2) The order of brackets

[[]]

"[" , "[" , "]" , "]"

Front

Back

Balanced!

[[][]]

0 1 2 3 4

1 0 2 3 0 4 0

Unbalanced!

[a][b]]

Ignore non "[" "]"

"]" ← remove a block

but I have no blocks!

][

isBalanced(queue)

[[]]

Balanced

[[][]

Not Balanced

[a][b]]

Not Balanced

][

Not Balanced

Block building

??

How do we know when a string is unbalanced?

↳ If we track order and count of "[", we can identify when there is no closed bracket

What values in my input queue do I need?

"[" "]"

How can I track these values?

leftRotateQueue

left

[a, b, c, d]

right

	rotation
[b, c, d, a]	1
[c, d, a, b]	2
[d, a, b, c]	3
[a, b, c, d]	4

x = [a, b, c, d]

c = x.dequeue() # c = a

x = [b, c, d]

x.enqueue(c)

[b, c, d, a]

leftRotateQueue

length 4

[a, b, c, d], 1 5 9

[b, c, d, a]

X = [a, b, c, d]

chasing x to store

[b, c, d, a]

How do we know which rotation we need?

X % length = rotation

What values in my input do I need to access?

↳ an offset # of values

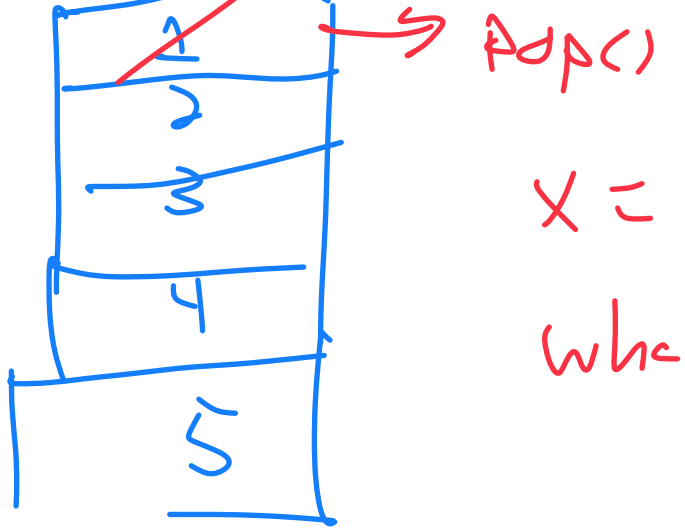
What do I do with every item I access?

↳ ???

removeOdds(stack)

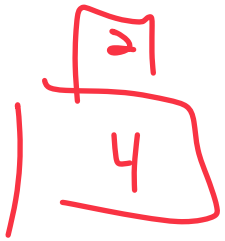
[~~1~~, 2, ~~3~~, 4, ~~5~~]

TOP



x = s.pop()

what do I do w/ x?



removeOdds(stack)

[1, 2, 3, 4, 5]

[2, 4]

[4, 2]

Can we remove some stack values and not others?

↳ No! We only have push() & pop()

What values in my input do I need to access?

↳ All of them

What do I do with every item I access?

↳ You decide

mergeSortedQueues

[1, 2, 3, 4, 5], [4, 5, 6, 7, 8]

New Queue

[1, 2, 3, 4, 4, 5, 5, 6, 7, 8]

mergeSortedQueues

5 5

[1, 2, 3, 4, 5], [4, 5, 6, 7, 8]

[1, 2, 3, 4, 4, 5, 5, 6, 7, 8]

What values in my input do I need to access?

↳ All items!

What do I do with every item I access?

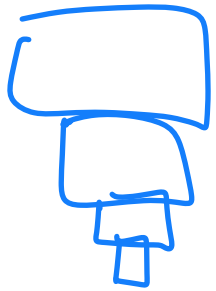
↳ Add to new queue!

How do I track order?

Do I need a new data structure?

reverseStack

[1, 2, 3, 4, 5]



reverseStack

[1, 2, 3, 4, 5]

[5, 4, 3, 2, 1]

Can I change the values in my stack directly?

↳ No!

What values in my input ~~queue~~ stack do I need?

↳ All of them

What should I do with the values I pop?