

Problem: Implement Queue using Stacks

Intro:

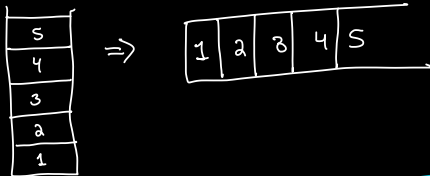
- Verify Constraints
- Create Testcases

Optimal:

- Brainstorming & Pattern Observations

Stack:

Queue:



push - adds to top

push - adds to end of queue

same `elements.push(x)`

peek/pop - look/remove top element

pop - removes first element
peek - looks at first element

use two stacks

`elements = [1, 2, 3, 4, 5]`

size

isEmpty

`save = []`

`while elements.size() > 0;`
`save.push(elements.pop())`

`pop()`
`result = save.pop()`

`peek()`
`result = save.peek()`

`while save.size() > 0:`
`elements.push(save.pop())`

`elements = [1]`
`save = [5, 4, 3, 2, 1]`

- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity
Time: push: $O(1)$, pop/peek: $O(n)$, empty: $O(1)$