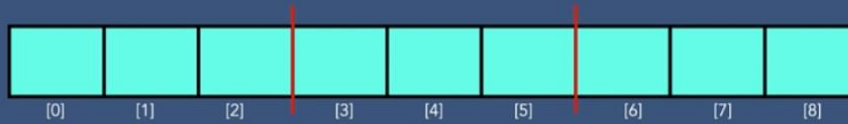


Three in One

Describe how you could use a single Array to implement three stacks.

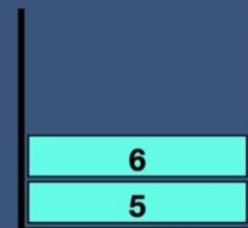


For Stack 1 — [0], [1], [2]	→	[0, n/3)
For Stack 2 — [3], [4], [5]	→	[n/3, 2n/3)
For Stack 3 — [6], [7], [8]	→	[2n/3, n)

Stack Min

How would you design a stack which, in addition to push and pop, has a function min which returns the minimum element? Push, pop and min should all operate in $O(1)$.

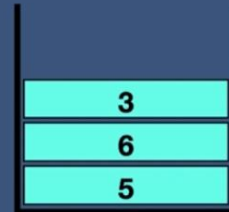
push(5)	min()	→ 5
push(6)	min()	→ 5
push(3)	min()	→ 3
push(7)	min()	→ 3
pop()	min()	→ 3
pop()	min()	→ 5



Stack Min

How would you design a stack which, in addition to push and pop, has a function min which returns the minimum element? Push, pop and min should all operate in $O(1)$.

push(5)	min() → 5
push(6)	min() → 5
push(3)	min() → 3
push(7)	min() → 3
pop()	min() → 3



Stack of Plates

Imagine a (literal) stack of plates. If the stack gets too high, it might topple. Therefore, in real life, we would likely start a new stack when the previous stack exceeds some threshold. Implement a data structure `SetOfStacks` that mimics this. `SetOfStacks.push()` and `SetOfStacks.pop()` should behave identically to a single stack (that is, `pop()` should return the same values as it would if there were just a single stack).

Follow Up:

Implement a function `popAt(int index)` which performs a pop operation on a specific sub - stack.

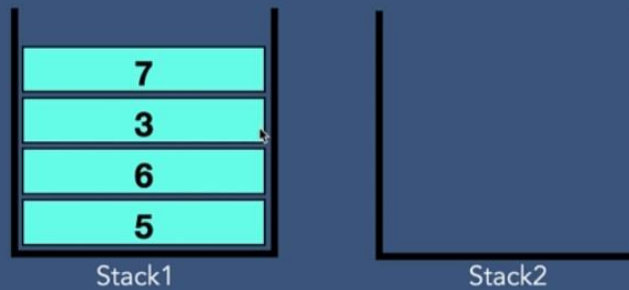


Threshold for the plates is five

Queue via Stacks

Implement Queue class which implements a queue using two stacks.

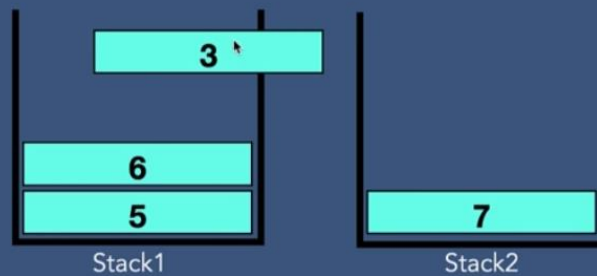
Enqueue()



Queue via Stacks

Implement Queue class which implements a queue using two stacks.

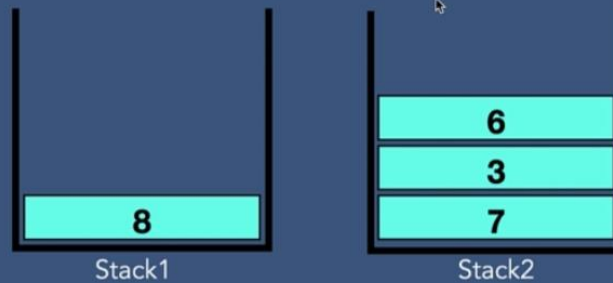
Dequeue()



Queue via Stacks

Implement Queue class which implements a queue using two stacks.

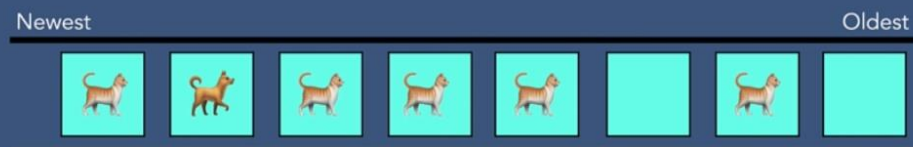
Dequeue()



Animal Shelter

An animal shelter, which holds only dogs and cats, operates on a strictly "first in, first out" basis. People must adopt either the "oldest" (based on arrival time) of all animals at the shelter, or they can select whether they would prefer a dog or a cat (and will receive the oldest animal of that type). They cannot select which specific animal they would like. Create the data structures to maintain this system and implement operations such as enqueue, dequeueAny, dequeueDog, and dequeueCat.

First In First Out



DequeueAny()

DequeueDog()