

Implementation

```
from stacks_and_queues import Stack
from linked_list import linked_list
from linked_list import Queue
```

```
class PseudoQueue():
    def __init__(self):
        self.in_stack = Stack()
        self.out_stack = Stack()
        self.front = None
    def enqueue(self, val):
        self.in_stack.push(val)
    def dequeue(self):
        if self.in_stack.is_empty():
            self.out_stack.pop()
            self.front = self.out_stack.top
        else:
            self.front = self.in_stack.pop()
    def peek(self):
        return self.front
```

Algorithm

Implementation 2 Stack objects.
Instantiate 2 stack objects.
Self are used for enqueue.
Stack two used for dequeue.
The enqueue method will add a value to
the rear of the queue. (stack one)
The dequeue method will pop each value
from the stack one pushing each value onto
stack two until the front of queue (stack one)
is found.

Iterate through input string list, enqueueing each
value onto stack one. When dequeue is
invoked, pop each value off stack one and
push each onto stack two until front of
stack one is found. That front value is popped
off stack one.

Problem Domain

Implement a queue using
two stacks.

input: [10, 15, 20], 5	enqueue
input: [5, 10, 15, 20]	dequeue
[5, 10, 15, 20]	input: [20]

Big O enqueue
space: O(1)
time: O(1)

Big O dequeue
space O(n)
time O(1)