Wk 2 Tutorial

Q1. Implement a stack using singly linked list.

Upon push, add element at the beginning of list.
(in front of HEAD)

Upon pop, remove element at beginning, set next element to HEAD, and return the removed element.

Flower Dance

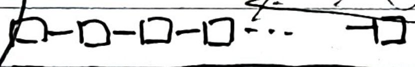
Flower

Q2. Implement a queue using singly linked list

Upon enqueue, add an element at the end of list. To keep this operation efficient, store a pointer to the last element of list.
Upon dequeue, remove an element at the beginning of the list

Q3. Traverse elements of singly linked list in reverse order.

(a) use O(1) space,



Best time complexity: $O(n^2)$

1 Start from HEAD and keep going next until there is no next. Print the element and

```
curr = None
last_element = None
while last_element ≠ HEAD:
    if curr.next == last_element:
        print(curr.value)
        last_element = curr
    curr = HEAD
```

(b) use $O(\sqrt{n})$ space

$\sqrt{n} \times \sqrt{n}$

floor($\sqrt{16}$)   ceil($\sqrt{16}$)

floor($\sqrt{n}$)

length

$\sqrt{3} \times \sqrt{3}$

floor(sqrt(49))

floor(sqrt(n))   prev pointer

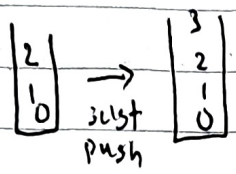
$O(n \cdot \sqrt{n})$

missed this → or $O(n)$ if use some additional space

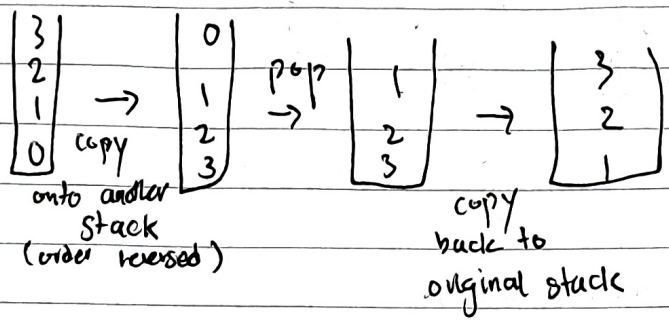
Q4

A) Implement Queue using stacks 기초임

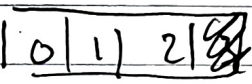
enqueue(3)



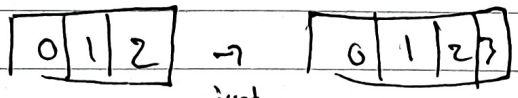
dequeue



B) Implement Stack by Queues

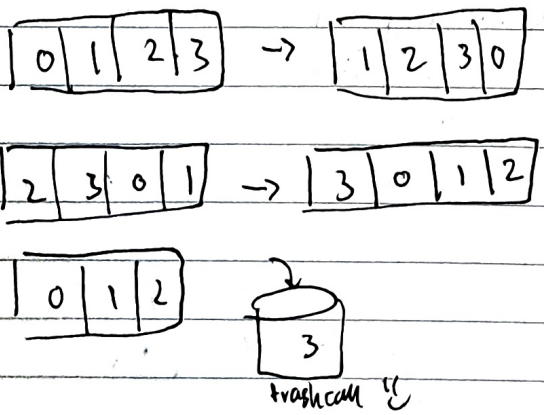


a) push(3)



just enqueue

b) pop()



n-1 steps.

} 1 step.