HW 3

Problem 1

a.

i. Stack1: 1 stack2: ø
ii. Stack1: 1, 2 stack2: ø
iii. Stack1: ø stack2: 2
iv. Stack1: 3 stack2: 2
v. Stack1: 3, 4 stack2: 2
vi. Stack1: 3, 4 stack2: ø
vii. Stack1: 3, 4, 5 stack2: ø
viii. Stack1: 3, 4, 5, 6 stack2: ø

b.

The worst case runtime of enqueue(X) is o(1).
The worst case runtime of dequeue() is o(n).

c.

- Dequeue
  - If stack 2 is not empty, the runtime will be O(1)
  - If stack 2 is empty, the runtime will be o(2n+1). One n for popping everything from stack1 and another n from pushing each element onto stack2
  - Thus the amortized runtime will be 2n+1/n = o(2n+1)/n)) which is approximately O(1)
- Enqueue:
  - Enqueue will always be o(1), so it's amortized runtime O(1)

d.

- Worst Case
  - Enqueue
    - Still o(1)
    - Amortized—> O(1/n)
  - Dequeue
    - The worst case scenario would occur when stack 2 has no elements. Since pop takes o(n) for each element of the stack, and then push takes 1 for each element of the stack, the worst runtime is O(n(n+1))—> O(n^2).
    - Over time, the amortized runtime is (n(n+1))/ n —> O(n+1)—> O(n)