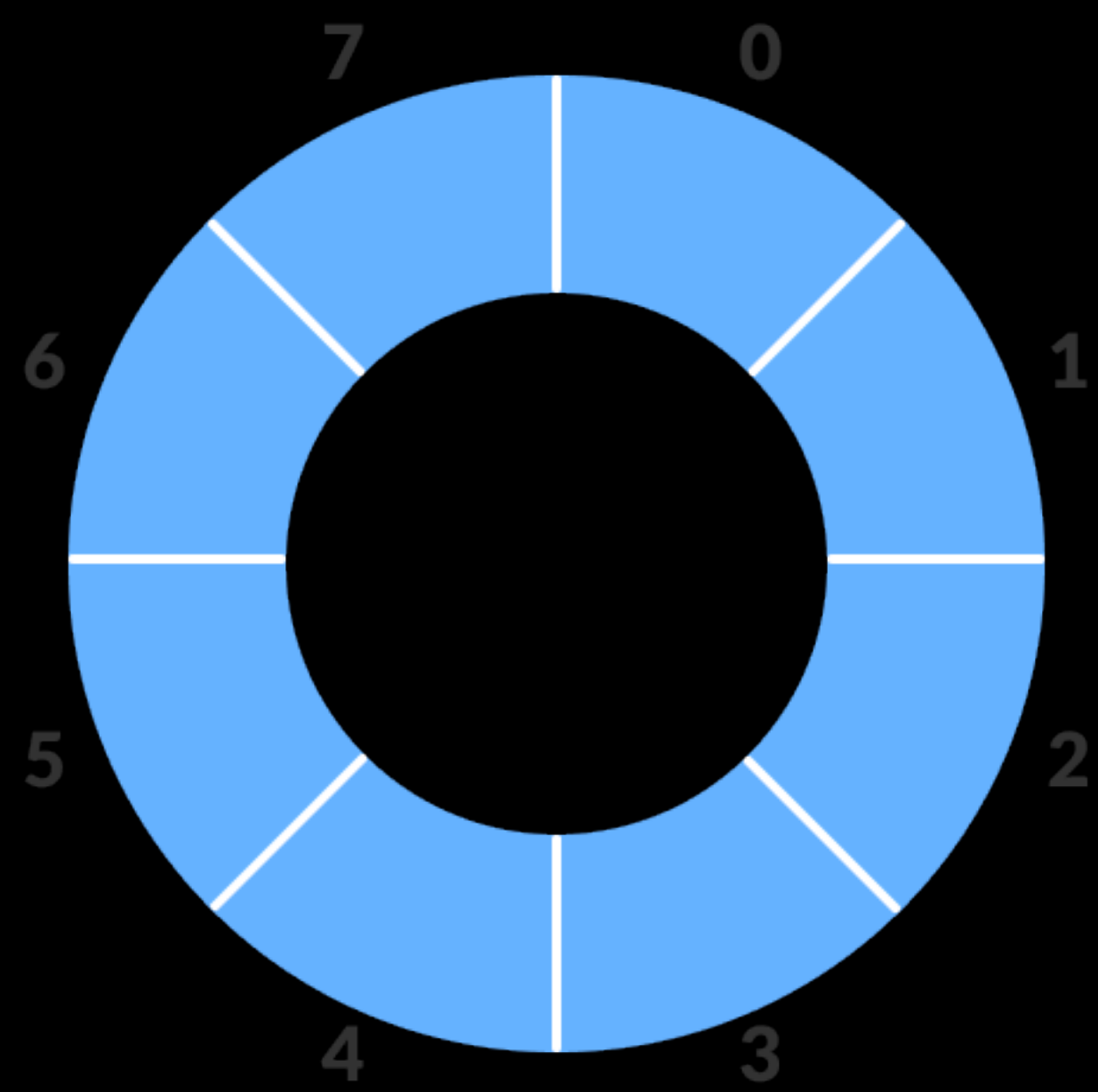


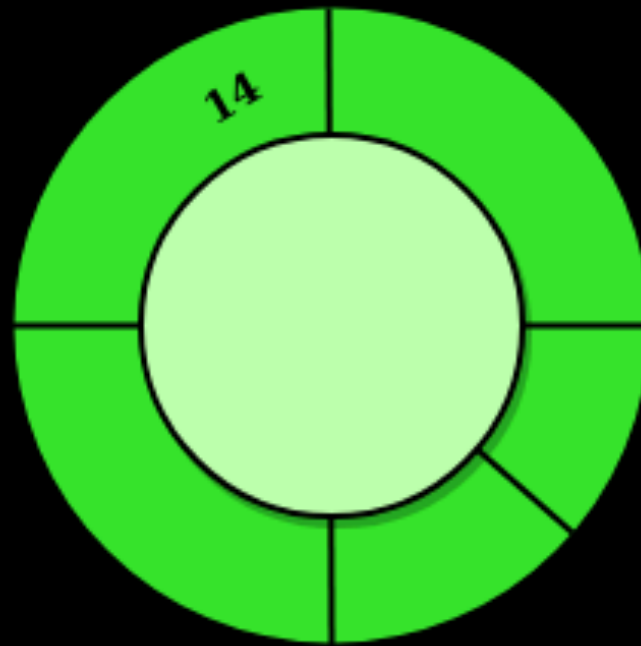
# 인공지능을 위한 알고리즘

# 원형 큐 (Circular Queue)

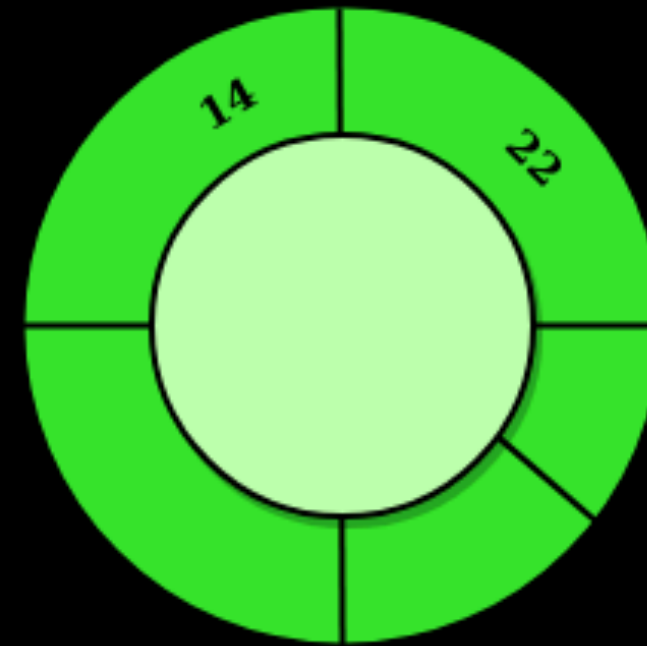




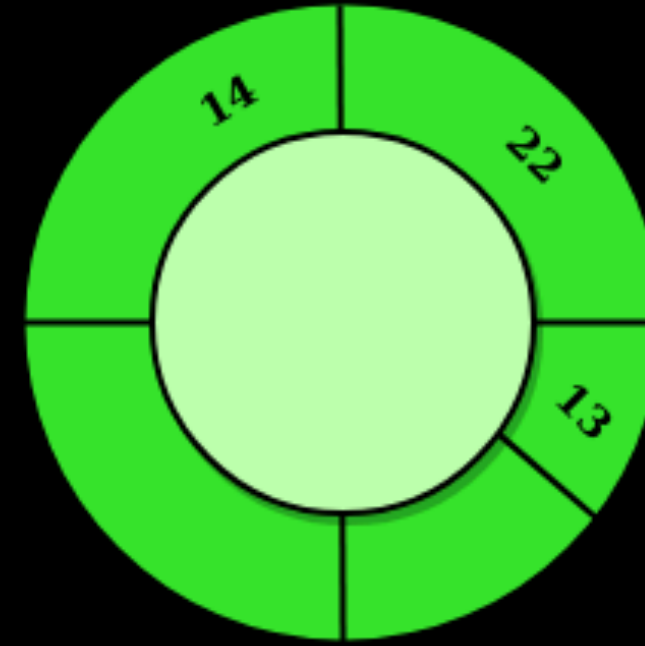
enqueue(14)



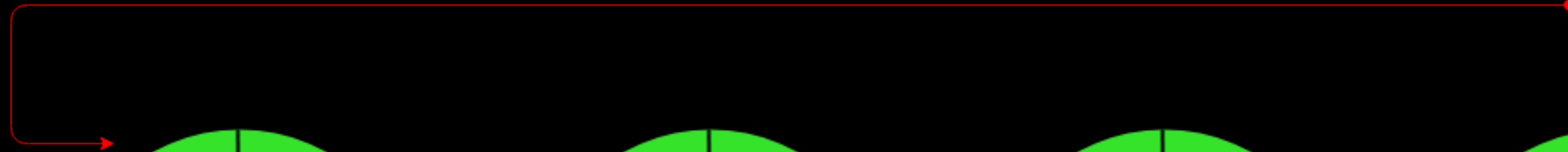
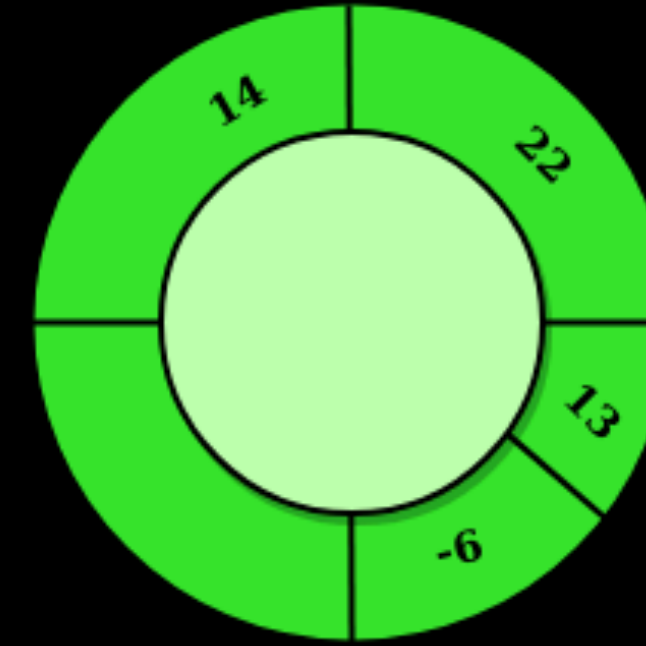
enqueue(22)



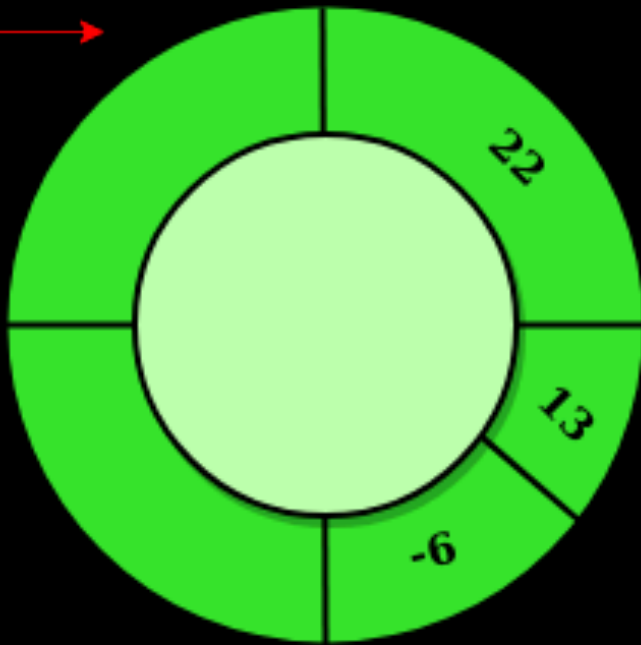
enqueue(13)



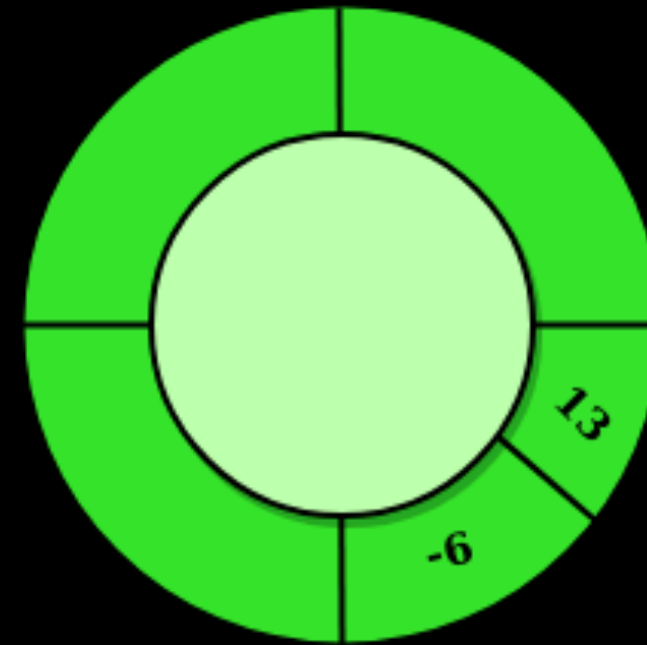
enqueue(-6)



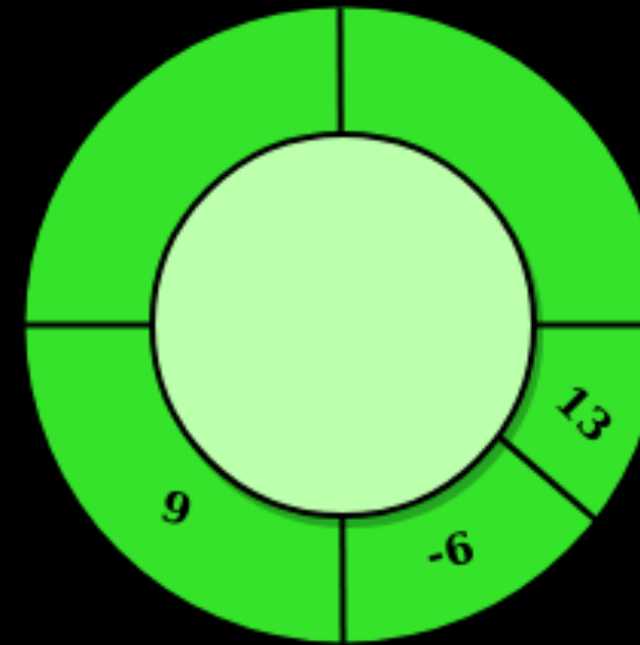
dequeue()



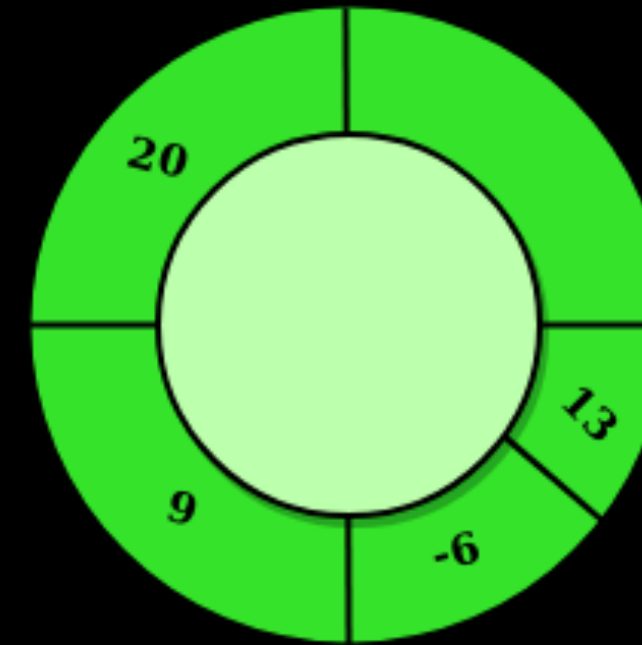
dequeue()



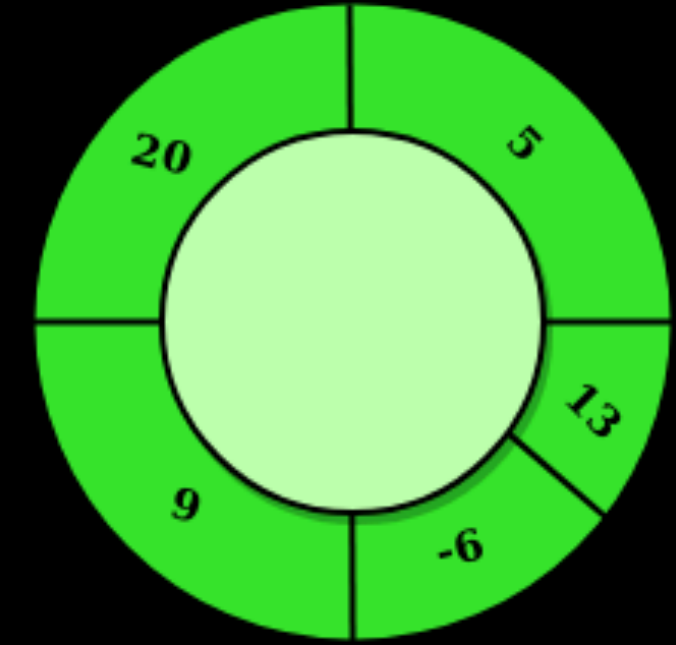
enqueue(9)



enqueue(20)



enqueue(5)



# 기존 큐를 원형 큐로 변경해보기

circular\_queue\_by\_class.py

- 첫 칸을 비우는 이유?

# 큐 추가 문제 (1) - 카드2

N장의 카드가 있다. 각각의 카드는 차례로 1부터 N까지의 번호가 붙어 있으며, 1번 카드가 제일 위에, N번 카드가 제일 아래인 상태로 순서대로 카드가 놓여 있다.

이제 다음과 같은 동작을 카드가 한 장 남을 때까지 반복하게 된다. 우선, 제일 위에 있는 카드를 바닥에 버린다. 그 다음, 제일 위에 있는 카드를 제일 아래에 있는 카드 밑으로 옮긴다.

# 큐 추가 문제 (1) - 카드2

예를 들어  $N=4$ 인 경우를 생각해 보자. 카드는 제일 위에서부터 1234 의 순서로 놓여 있다. 1을 버리면 234가 남는다. 여기서 2를 제일 아래로 옮기면 342가 된다. 3을 버리면 42가 되고, 4를 밑으로 옮기면 24가 된다. 마지막으로 2를 버리고 나면, 남는 카드는 4가 된다.

$N$ 이 주어졌을 때,

제일 마지막에 남게 되는 카드를 구하는 프로그램을 작성하시오.



# 큐 추가 문제 (1) - 카드2

- 첫째 줄에 정수  $N$  ( $1 \leq N \leq 500,000$ )이 주어진다.
- 첫째 줄에 남게 되는 카드의 번호를 출력한다.

# 큐 추가 문제 (1) - 카드2

- 입력 예제 1

6

- 출력 예제 1

4

card2\_deque.py

# 큐 추가 문제 (2) - 회전하는 큐

지민이는  $N$ 개의 원소를 포함하고 있는 양방향 순환 큐를 가지고 있다. 지민이는 이 큐에서 몇 개의 원소를 뽑아내려고 한다.

지민이는 이 큐에서 다음과 같은 3가지 연산을 수행할 수 있다.

# 큐 추가 문제 (2) - 회전하는 큐

첫 번째 원소를 뽑아낸다. 이 연산을 수행하면, 원래 큐의 원소가  $a_1, \dots, a_k$ 이었던 것이  $a_2, \dots, a_k$ 와 같이 된다.

왼쪽으로 한 칸 이동시킨다. 이 연산을 수행하면,  $a_1, \dots, a_k$ 가  $a_2, \dots, a_k, a_1$ 이 된다.

오른쪽으로 한 칸 이동시킨다. 이 연산을 수행하면,  $a_1, \dots, a_k$ 가  $a_k, a_1, \dots, a_{k-1}$ 이 된다.

# 큐 추가 문제 (2) - 회전하는 큐

큐에 처음에 포함되어 있던 수  $N$ 이 주어진다. 그리고 지민이가 뽑아내려고 하는 원소의 위치가 주어진다. (이 위치는 가장 처음 큐에서의 위치이다.)

이때, 그 원소를 주어진 순서대로 뽑아내는데 드는 2번, 3번 연산의 최소값을 출력하는 프로그램을 작성하시오.



# 큐 추가 문제 (2) - 회전하는 큐

- 첫째 줄에 큐의 크기  $N$ 과 뽑아내려고 하는 수의 개수  $M$ 이 주어진다.
- $N$ 은 50보다 작거나 같은 자연수이고,  $M$ 은  $N$ 보다 작거나 같은 자연수이다.
- 둘째 줄에는 지민이가 뽑아내려고 하는 수의 위치가 순서대로 주어진다. 위치는 1보다 크거나 같고,  $N$ 보다 작거나 같은 자연수이다.

# 큐 추가 문제 (2) - 회전하는 큐

- 입력 예제 1

10 3

1 2 3

- 출력 예제 1

0

# 큐 추가 문제 (2) - 회전하는 큐

- 입력 예제 2

10 3

2 9 5

- 출력 예제 2

8