10월 6일 5주차 수업 리포트

1. **원형큐(circular queue) 프로그램을 덱(double-ended queue) 버전으로 확장**

* Deque 이라는 class를 먼저 만든 다음, 그 아래에 다음과 같은 기능을 집어넣었다. Deque는 앞쪽과 뒤쪽 모든 방향에서 데이터를 삽입, 추출, 확인할 수 있는 자료구조이기 때문에 add, delete, get을 앞, 뒤에서 모두 가능하도록 만들었다.
* 인덱스가 배열의 끝에 도달하면 ‘%’ 연산자를 이용해서 0이 되도록 만들었다.
  + Front에서는 self.front = (self.front + 1) % MAX\_QUEUE\_SIZE의 형태로 나타내었다.
  + Rear에서는 self.rear = (self.rear – 1 + MAX\_QUEUE\_SIZE) % MAX\_QUEUE\_SIZE의 형태로 나타내었다.
* **소스 코드**

MAX\_QUEUE\_SIZE = 10

class Deque:

def \_\_init\_\_(self):

self.queue = [None for \_ in range(MAX\_QUEUE\_SIZE)]

self.front = 0

self.rear = 0

def is\_empty(self):

return self.front == self.rear

def is\_full(self):

return (self.rear + 1) % MAX\_QUEUE\_SIZE == self.front

def add\_front(self, item):

if self.is\_full():

print("Deque이 꽉 찼습니다.")

return

self.front = (self.front - 1 + MAX\_QUEUE\_SIZE) % MAX\_QUEUE\_SIZE

self.queue[self.front] = item

def delete\_front(self):

if self.is\_empty():

print("Deque이 비었습니다.")

return None

item = self.queue[self.front]

self.front = (self.front + 1) % MAX\_QUEUE\_SIZE

return item

def get\_front(self):

if self.is\_empty():

print("Deque이 비었습니다.")

return None

return self.queue[self.front]

def add\_rear(self, item):

if self.is\_full():

print("Deque이 꽉 찼습니다.")

return

self.queue[self.rear] = item

self.rear = (self.rear + 1) % MAX\_QUEUE\_SIZE

def delete\_rear(self):

if self.is\_empty():

print("Deque이 비었습니다.")

return None

self.rear = (self.rear - 1 + MAX\_QUEUE\_SIZE) % MAX\_QUEUE\_SIZE

return self.queue[self.rear]

def get\_rear(self):

if self.is\_empty():

print("Deque이 비었습니다.")

return None

return self.queue[(self.rear - 1 + MAX\_QUEUE\_SIZE) % MAX\_QUEUE\_SIZE]

def size(self):

return (self.rear - self.front + MAX\_QUEUE\_SIZE) % MAX\_QUEUE\_SIZE

def menu():

dq = Deque()

while True:

operation = input("앞쪽에서 삽입(I)/앞쪽에서 추출(D)/앞쪽에서 확인(P)/뒤쪽에서 삽입(E)/뒤쪽에서 추출(R)/뒤쪽에서 확인(G)/크기(S)/종료(Q) 중 하나를 선택 ==> ").upper()

if operation == 'I':

item = int(input("앞쪽에 입력할 데이터 ==> "))

dq.add\_front(item)

elif operation == 'D':

dq.delete\_front()

elif operation == 'P':

print(dq.get\_front())

elif operation == 'E':

item = int(input("뒤쪽에 입력할 데이터 ==> "))

dq.add\_rear(item)

elif operation == 'R':

dq.delete\_rear()

elif operation == 'G':

print(dq.get\_rear())

elif operation == 'S':

print(dq.size())

elif operation == 'Q':

break

else:

print("입력이 잘못됨")

if \_\_name\_\_ == "\_\_main\_\_":

menu()

* **수행 결과**

