

자료구조 5장 과제

Global Business & Technology

201904385 우인경

1. 원형 큐 구현 및 실습

<코드 화면>

```
MAX_QSIZE = 10

class CircularQueue:
    def __init__(self):
        self.front = 0
        self.rear = 0
        self.items = [None] * MAX_QSIZE
    def isEmpty(self):
        return self.front == self.rear
    def isFull(self):
        return self.front == (self.rear+1) % MAX_QSIZE
    def claer(self):
        self.front = self.rear
    def enqueue(self, item):
        if not self.isFull():
            self.rear = (self.rear+1) % MAX_QSIZE
            self.items[self.rear] = item
    def dequeue(self):
        if not self.isEmpty():
            self.front = (self.front+1) % MAX_QSIZE
            return self.items[self.front]
    def peek(self):
        if not self.isEmpty():
            return self.items[(self.front + 1) % MAX_QSIZE]
    def size(self):
        return (self.rear - self.front + MAX_QSIZE) % MAX_QSIZE
    def display(self):
        out = []
        if self.front < self.rear:
            out = self.items[self.front+1 : self.rear+1 ]
        else:
            out = self.items[self.front+1:MAX_QSIZE] + self.items[0:self.rear+1]
        print("[f=%s, r=%d] ==>" % (self.front, self.rear), out)

q = CircularQueue()

for i in range(8):
    q.enqueue(i)
q.display()

for i in range(5):
    q.dequeue()
q.display()

for i in range(8, 14):
    q.enqueue(i)
q.display()
```

<실행 화면>

```
===== RESTART: /Users/inkyung/Documents/
[f=0, r=8] ==> [0, 1, 2, 3, 4, 5, 6, 7]
[f=5, r=8] ==> [5, 6, 7]
[f=5, r=4] ==> [5, 6, 7, 8, 9, 10, 11, 12, 13]
>>>
```

2. 미로탐색 구현 및 실습

<코드 화면>

```
from CircularQueue import *

def isValidPos(x,y):
    if x < 0 or y < 0 or x >= MAX_SIZE or y >= MAX_SIZE:
        return False
    else:
        return map[x][y] == '0' or map[x][y] == 'x'

def BFS():
    que = CircularQueue()
    que.enqueue( (1,0) )
    print('BFS: ')

    while not que.isEmpty():
        here = que.dequeue()
        print(here, end='-->')
        x, y = here
        if map[x][y] == 'x':
            return True
        else:
            map[x][y] = '.'
            if isValidPos(x-1, y): que.enqueue( (x-1,y) ) #상
            if isValidPos(x+1, y): que.enqueue( (x+1,y) ) #하
            if isValidPos(x, y-1): que.enqueue( (x,y-1) ) #좌
            if isValidPos(x, y+1): que.enqueue( (x,y+1) ) #우

    return False

map = [ ['1', '1', '1', '1', '1', '1'],
        ['e', '0', '1', '0', '0', '1'],
        ['1', '0', '0', '0', '1', '1'],
        ['1', '0', '1', '0', '1', '1'],
        ['1', '0', '1', '0', '0', 'x'],
        ['1', '1', '1', '1', '1', '1'] ]

MAX_SIZE = 6
result = BFS()

if result:
    print(' --> 미로탐색 성공')
else:
    print(' --> 미로탐색 실패')
```

<실행 화면>

```
===== RESTART: /Users/inkyung/Documents/Queue_maze.py
=====
BFS:
(1, 0)-->(1, 1)-->(2, 1)-->(3, 1)-->(2, 2)-->(4, 1)-->(2, 3)-->(1, 3)-->
(3, 3)-->(1, 4)-->(4, 3)-->(4, 4)-->(4, 5)--> --> 미로탐색 성공
>>> |
```

3. 우선순위 큐를 이용한 미로탐색 구현 및 실습

<코드 화면>

```
class PriorityQueue:

    def __init__(self):
        self.items = []

    def isEmpty(self):
        return len(self.items) == 0
    def size(self):
        return len(self.items)
    def clear(self):
        self.items = []

    def enqueue(self, item):
        self.items.append(item)

    def findMaxIndex(self):
        if self.isEmpty():
            return None
        else:
            highest = 0
            for i in range(1, self.size()):
                if self.items[i] > self.items[highest]:
                    highest = i
            return highest

    def dequeue(self):
        highest = self.findMaxIndex()
        if highest is not None:
            return self.items.pop(highest)

    def peeks(self):
        highest = findMaxIndex()
        if highest is not None:
            return self.items[highest]

q = PriorityQueue()

q.enqueue(34)
q.enqueue(18)
q.enqueue(27)
q.enqueue(45)
q.enqueue(15)

print("PQueue: ", q.items)

while not q.isEmpty():
    print("Max Priority = ", q.dequeue() )
```

<실행 화면>

```
===== RESTART: /Users/inkyung/Documents/Priority_Queue.py =====
PQueue: [34, 18, 27, 45, 15]
Max Priority = 45
Max Priority = 34
Max Priority = 27
Max Priority = 18
Max Priority = 15
>>> |
```

4. 피보나치 수열

<코드 화면>

```
from CircularQueue import *

q = CircularQueue()
q.enqueue(0)
q.enqueue(1)

print("피보나치 수열: ", 0, 1, end=' ')

for i in range(2,10):
    num1 = q.dequeue()
    num2 = q.peek()
    q.enqueue(num1 + num2)
    print(num1+num2, end=' ')
```

<실행 화면>

```
===== RESTART: /Users/inkyung/
피보나치 수열:  0 1 1 2 3 5 8 13 21 34
>>> |
```