

| 연습문제 |

5.1 다음 중 큐에 대한 설명이 아닌 것은?

- ① 입력과 출력이 각각 후단과 전단으로 제한된다.
- ② 함수 호출시 복귀 주소를 저장하기 위해 사용된다.
- ③ FIFO(First-In First-Out) 방식으로 동작한다.
- ④ 배열 구조와 연결된 구조로 구현이 가능하다.

5.2 다음 중 자료구조에서 큐(Queue)를 활용하는 분야로 거리가 먼 것은?

- ① 이진 트리의 레벨 순회
- ② 그래프의 탐색에서 너비우선탐색
- ③ 기수 정렬
- ④ 집합의 구현

5.3 스택을 이용해서 큐를 구현할 수 있다. 몇 개의 스택이 필요할까? 리스트와 같은 다른 자료 구조는 없다고 가정하라.

- ① 1
- ② 2
- ③ 3
- ④ 4

5.4 원형 큐의 front 와 rear 의 값이 각각 7과 2일 때, 이 원형 큐가 가지고 있는 데이터의 개수는? (단, 배열의 크기 capacity는 12이고, front와 rear의 초기값은 0이다.) 17개 8, 9, 10, 11, 12, 5

5.5 다음 중 원형 큐에서 공백상태를 검사할 때 사용하는 조건은? (단, capacity는 원형 큐를 위한 배열의 크기이다.)

- ① $front == 0$ and $rear == 0$
- ② $front == (capacity-1)$ and $rear == (capacity-1)$
- ③ $front == rear$
- ④ $front == (rear+1) \% capacity$

5.6 위 문제에서 원형 큐의 포화상태를 검사할 때 사용하는 조건은?

5.7 그림과 같은 원형 큐에서 다음의 연산을 차례로 수행한다고 하자. 각 단계가 수행될 때 큐의 상태를 그려라. 현재 front는 0이고 rear는 2라고 하자.

[0]	[1]	[2]	[3]	[4]
	B	C		

- (1) A 추가 (2) D 추가 (3) 삭제
(4) E 추가 (5) 삭제

5.8 크기가 5인 공백상태의 원형 큐에 다음 연산들이 순서대로 실행되었다. 각 단계에서의 원형 큐의 내용(배열의 내용과 front와 rear의 값)을 나타내라. 단, 맨 처음에 front와 rear는 모두 0이라고 가정하라.

enqueue(1), enqueue(2), enqueue(3), dequeue(), enqueue(4)

5.9 원형 큐에 항목들을 삽입하고 삭제하는 연산은 시간 복잡도가 어떻게 되는가?

- ① $O(1)$ ② $O(\log_2 n)$ ③ $O(n)$ ④ $O(n^2)$

5.10 다음 코드의 연산 결과 큐에 남아 있는 내용을 순서대로 적어라.

```
values = Queue()
for i in range( 20 ) :
    if i % 3 == 0 :
        values.enqueue( i )
```

0, 3, 6, 9, 12, 15, 18

```
values = Queue()
for i in range( 20 ) :
    if i % 3 == 0 :
        values.enqueue( i )
    elif i % 4 == 0 :
        values.dequeue()
```

0, 3, 6, 9, 12, 15, 18

5.11 텍에 대하여 올바른 설명을 모두 골라라.

- ① 양쪽 끝에서 추가와 삭제가 가능하다.
② 배열로 구현될 수 없다.
③ 선형 자료구조이다.
④ 삽입된 순서와 상관없이 우선순위가 높은 항목이 먼저 출력된다.

5.8

~~front~~

enqueue(1)

Front = 0 Rear = 0



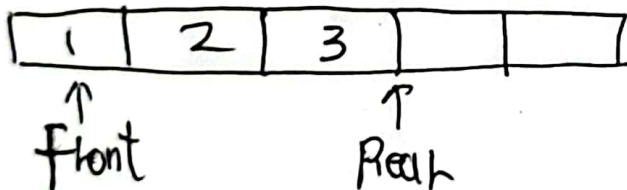
enqueue(2)

Front = 0 Rear = 1



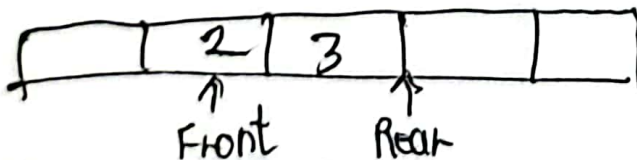
enqueue(3)

Front = 0 Rear = 2



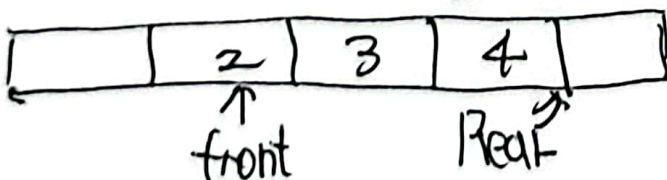
dequeue(4)

Front = 1 Rear = 2



enqueue(4)

Front = 1 Rear = 3





5.12 다음은 원형 덱의 연산들이다. front나 rear가 증가되어 인덱스를 시계방향으로 회전시켜야 하는 연산들을 모두 골라라. 2, 4, 5

- ☐ ① isEmpty() ☒ ② isFull() ☒ ③ addFront(x) ☒ ④ deleteFront()
☒ ⑤ getFront() ☒ ⑥ addRear(x) ☒ ⑦ deleteRear() ☒ ⑧ getRear()

5.13 위 문제에서 인덱스가 감소되어 반대방향(반시계방향)으로 회전시켜야 하는 연산들을 모두 골라라.

5.14 원형 덱에서 front를 시계방향과 반시계방향으로 회전시킬 수 있는 방법을 유사 코드(또는 파이썬 코드)로 적어라.

5.15 덱 D에 숫자 (1,2,3,4,5,6,7,8)가 순서대로 저장되어 있다. 공백상태의 큐인 Q가 있을 때 덱 D의 내용을 (8,7,6,5,4,3,2,1) 으로 바꾸는 코드를 작성하라. 단, D와 Q이외의 다른 변수는 사용할 수 없다.

5.16 위의 문제에서 이번에는 큐가 아니라 스택 S를 사용하여 동일한 결과가 나타나도록 코드를 수정하라.

5.17 우선순위 큐에 대한 설명으로 거리가 먼 것은? (2)

- ☐ ① 우선순위의 개념을 큐에 도입한 자료구조
☒ ② 선형 자료구조
☐ ③ 입력 순서와 상관없이 우선순위가 높은 데이터가 먼저 출력
☒ ④ 가장 일반적인 큐

5.18 우선순위 큐를 선형 자료구조로 볼 수 없는 이유를 설명하라.

5.19* 스택이나 큐를 직접 구현할 필요 없이 파이썬의 큐 모듈을 이용해 미로탐색 프로그램을 구현할 수 있다. 5.3절의 코드를 수정하여 파이썬의 큐 모듈을 이용한 깊이우선탐색과 너비우선탐색 함수를 구현하라.