

CHAPTER.13

큐의 기본 및 간단 구현





학습 내용

[1] 큐의 기본

[2] 큐의 간단 구현



학습 목표

- ④ 큐의 개념을 설명할 수 있다.
- ④ 큐에 데이터를 넣거나 추출하는 원리를 설명할 수 있다.



01

큐의 기본

- 1) 큐란?
- 2) 큐의 개념
- 3) 큐의 원리

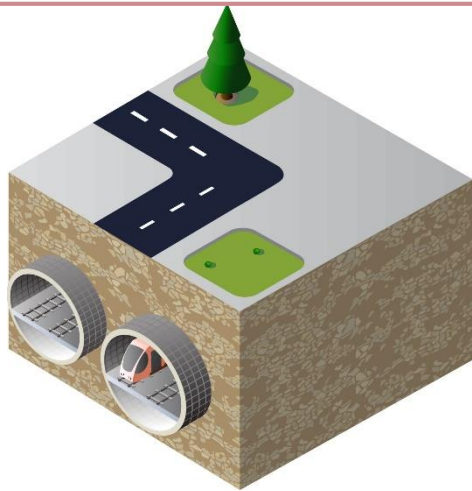


1] 큐란?



큐(Queue)

먼저 들어간 것이 먼저 나오는 구조를 의미



기차가 터널에 들어가는
순서대로 터널을 빠져나옴



ATM기에서 줄을 선 순서대로
예금을 인출

2] 큐의 개념

큐의 기본 구조

큐(Queue) 자료구조는 입구와 출구가 따로 있는 원통 형태 (FIFO)

■ 스택과 큐의 차이점

입구 겸 출구



스택



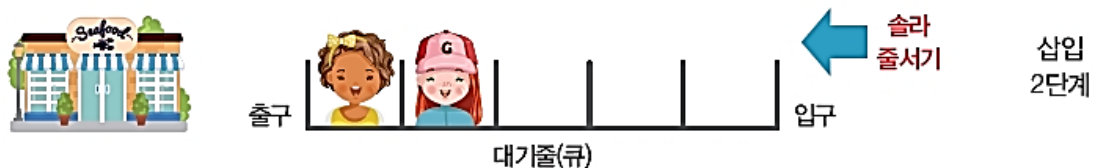
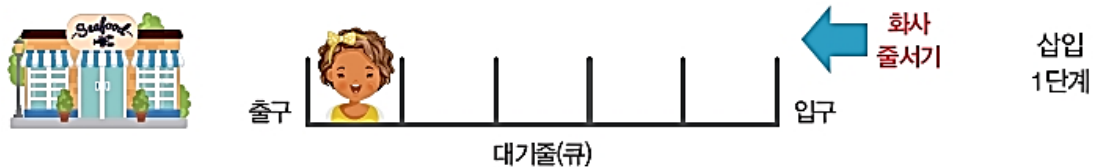
큐

2] 큐의 개념



큐의 기본 구조

대기줄에 차례대로 줄을 서는 예



2] 큐의 개념

큐의 기본 구조

대기줄에 차례대로 나오는 예



대기 손님
나오기



출구



입구

추출
1단계

대기줄(큐)



대기 손님
나오기



출구



입구

추출
2단계

대기줄(큐)



대기 손님
나오기



출구



입구

추출
3단계

대기줄(큐)

3] 큐의 원리



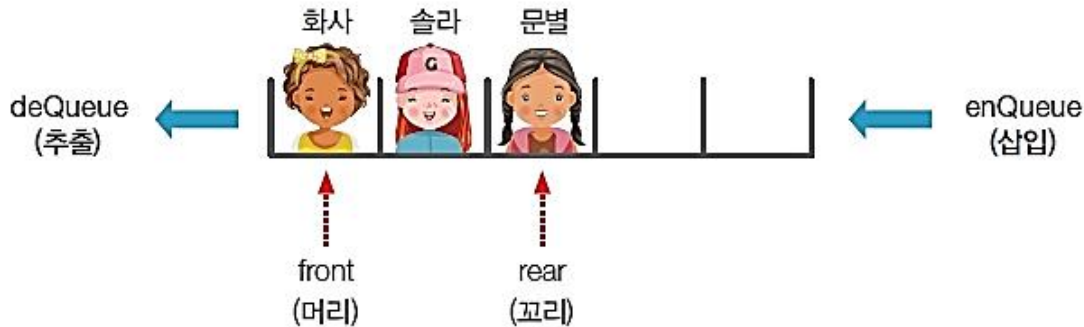
큐의 구조와 용어

큐에 데이터를 **삽입**하는 작동 : enqueue(인큐)

데이터를 **추출**하는 작동 : dequeue(데큐)

저장된 데이터 중 **첫 번째** 데이터 : front(머리)

저장된 데이터 중 **마지막** 데이터 : rear(꼬리)



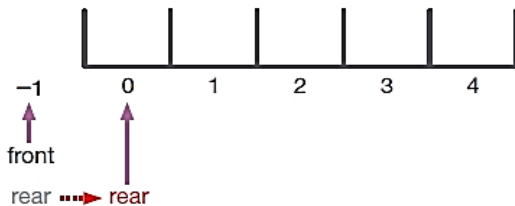
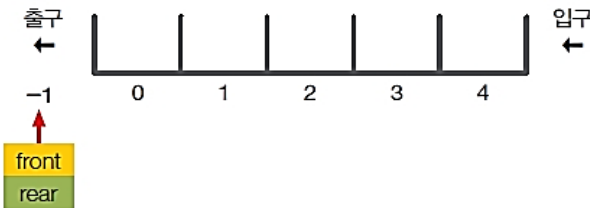
3] 큐의 원리

데이터 삽입 : enqueue

큐에서 데이터를 삽입(enQueue)하는 과정

1단계

- rear를 한칸 오른쪽으로 이동



2단계

- rear 위치에 '화사'를 입력(enQueue)



3] 큐의 원리



데이터 추출 : deQueue

큐에서 데이터를 추출(deQueue)하는 과정

1단계

- front를 한칸
오른쪽으로 이동



2단계

- front 위치의
'화사'를 추출
(deQueue)





02

큐의 간단 구현

- 1) 큐 생성
- 2) 데이터 삽입 : enQueue
- 3) 데이터 추출 : deQueue
- 4) 큐의 간단 구현 실습

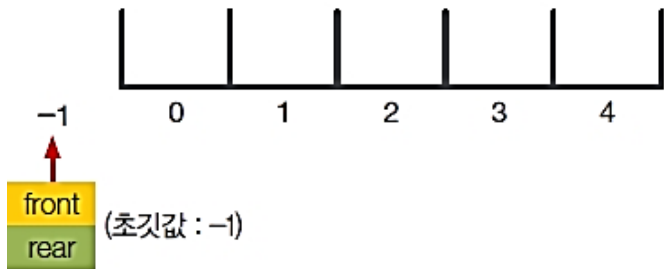


배열 크기를 지정한 후 해당 크기의 빈 큐 생성

▪ 크기가 5칸인 큐의 초기 상태

```
queue = [None, None, None, None, None]
```

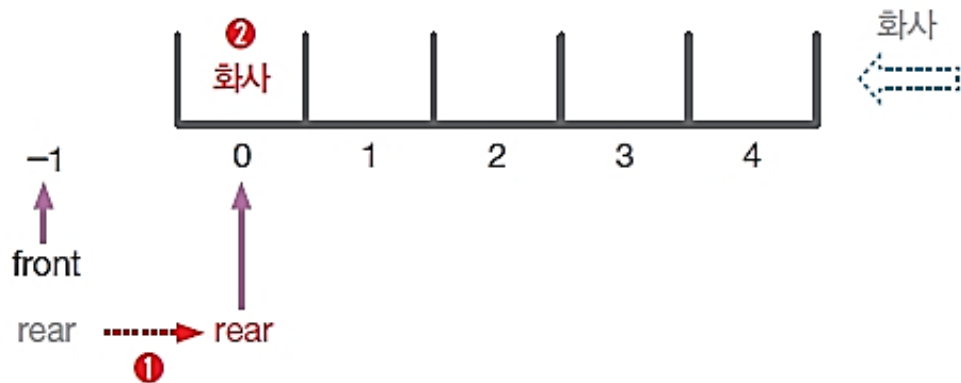
```
front = rear = -1
```



2] 데이터 삽입 : enqueue



큐의 데이터 삽입 과정



큐의 데이터 삽입 과정

- 크기가 5칸인 큐의 생성과 데이터 3개 입력

```
1 queue = [None, None, None, None, None]
2 front = rear = -1
3
4 rear += 1 ❶
5 queue[rear] = "화사" ❷
6 rear += 1
7 queue[rear] = "솔라"
8 rear += 1
9 queue[rear] = "문별"
10
11 print("----- 큐 상태 -----")
12 print('[출구] <-- ', end = ' ')
```

큐의 데이터 삽입 과정

- 크기가 5칸인 큐의 생성과 데이터 3개 입력

```
13 for i in range(0, len(queue), 1) :  
14     print(queue[i], end = ' ')  
15 print('<-- [입구]')
```

실행 결과

----- 큐 상태 -----

[출구] <-- 화사 솔라 문별 None None <-- [입구]

3] 데이터 추출 : deQueue



큐의 데이터 추출 과정



큐의 데이터 추출 과정

- 큐에서 데이터 3개 추출

```
1 queue = ["화사", "솔라", "문별", None, None]
2 front = -1
3 rear = 2
4
5 print("----- 큐 상태 -----")
6 print('[출구] <-- ', end = ' ')
7 for i in range(0, len(queue), 1):
8     print(queue[i], end = ' ')
9 print('<-- [입구]')
10 print("-----")
11
```

큐의 데이터 추출 과정

- 큐에서 데이터 3개 추출

```
12 front += 1
13 data = queue[front]
14 queue[front] = None
15 print('deQueue --> ', data)
16
17 front += 1
18 data = queue[front]
19 queue[front] = None
20 print('deQueue --> ', data)
21
22 front += 1
23 data = queue[front]
```

큐의 데이터 추출 과정

- 큐에서 데이터 3개 추출

```
24 queue[front] = None
25 print('deQueue --> ', data)
26 print("-----")
27
28 print("----- 큐 상태 -----")
29 print('[출구] <-- ', end = ' ')
30 for i in range(0, len(queue), 1) :
31     print(queue[i], end = ' ')
32 print('<-- [입구]')
```

큐의 데이터 추출 과정

- 큐에서 데이터 3개 추출

실행 결과

----- 큐 상태 -----

[출구] <-- 화사 솔라 문별 None None <-- [입구]

deQueue --> 화사

deQueue --> 솔라

deQueue --> 문별

----- 큐 상태 -----

[출구] <-- None None None None None <-- [입구]

4] 큐의 간단 구현 실습



큐의 간단 구현 실습



Q1

Q2

Q3

Q1

대기줄(큐)에 줄을 서는 동작과 거리가 먼 것은?

- 1 줄을 먼저 선 사람이 가장 먼저 입장한다.
- 2 입구와 출구가 따로 있다.
- 3 출구는 한곳으로만 사용한다.
- 4 필요할 경우 가장 나중에 줄을 선 사람이 가장 먼저 입장되기도 한다.

Q1

Q2

Q3

Q1

대기줄(큐)에 줄을 서는 동작과 거리가 먼 것은?

1 줄을 먼저 선 사람이 가장 먼저 입장한다.

2 입구와 출구가 따로 있다.

3 출구는 한곳으로만 사용한다.



필요할 경우 가장 나중에 줄을 선 사람이 가장 먼저 입장되기도 한다.

정답

4 필요할 경우 가장 나중에 줄을 선 사람이 가장 먼저 입장되기도 한다.

해설

큐는 줄을 먼저 선 사람이 먼저 들어가는 FIFO 구조입니다.

Q1

Q2

Q3

Q2

다음 중 큐에서 삽입과 추출에 사용되는 용어 네 가지를
고르시오.

insert, rear, delete, top, enqueue, bottom,
front, header, current, pre, pop, dequeue

Q1

Q2

Q3

Q2

다음 중 큐에서 삽입과 추출에 사용되는 용어 네 가지를 고르시오.

insert, rear, delete, top, enqueue, bottom,
front, header, current, pre, pop, dequeue

enqueue

dequeue

front

rear

정답

enqueue, dequeue, front, rear

해설

큐에 데이터를 삽입하는 작동은 enqueue(인큐), 데이터를 추출하는 작동은 dequeue(데큐)이며, 저장된 데이터 중 첫 번째 데이터를 front(머리), 저장된 데이터 중 마지막 데이터를 rear(꼬리)라고 합니다.

Q1

Q2

Q3

Q3

큐에서 (1)은 데이터를 삽입하는 코드이다.
front와 rear와 관련된 코드이다. (1)을 채우시오.

```
queue = [None, None, None, None, None]
front = rear = -1
```

```
(      1      )
queue[rear] = "화사"
```

(1)

Q1

Q2

Q3

Q3

큐에서 (1)은 데이터를 삽입하는 코드이다.
front와 rear와 관련된 코드이다. (1)을 채우시오.

```
queue = [None, None, None, None, None]
front = rear = -1
```

```
(    1    )
queue[rear] = "화사"
```

(1) rear += 1

정답

rear += 1

해설

rear를 증가시키고 해당 위치에 데이터를 삽입합니다.

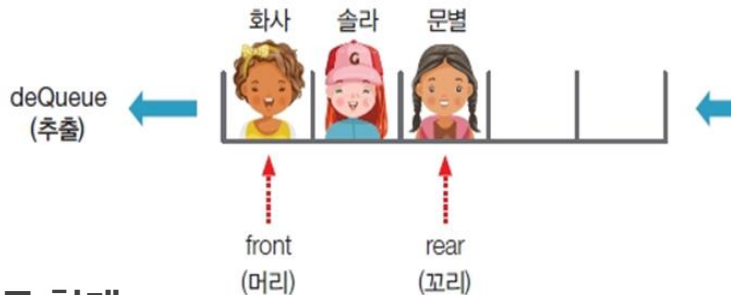
큐의 기본

④ 큐의 개념

- 큐(Queue) 자료구조는 입구와 출구가 따로 있는 원통 형태

④ 큐의 구조와 용어

- 큐에 데이터를 삽입하는 작동 : enQueue(인큐)
- 데이터를 추출하는 작동 : deQueue(데큐)
- 저장된 데이터 중 첫 번째 데이터 : front(머리)
- 저장된 데이터 중 마지막 데이터 : rear(꼬리)



큐의 간단 구현

④ 큐 생성

- 배열 크기를 지정한 후 해당 크기의 빈 큐 생성

```
queue = [None, None, None, None, None]
```

```
front = rear = -1
```

