



동아대학교

# 데이터 구조

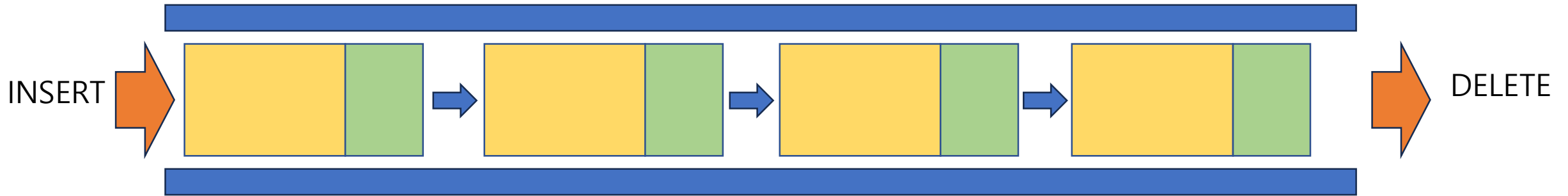
6주차: 큐

박 경 훈

# QUEUE

- 큐는 한쪽 끝에서만 삽입이 이루어지고 다른 한쪽 끝에서는 삭제 연산만 이루어지는 유한 순서 리스트

FIFO (First In First Out)



- 큐의 주요 동작(메소드)

- enqueue() : 큐에 데이터를 넣는다.
- dequeue() : 큐에서 데이터를 빼낸다.
- isEmpty() : 큐가 비어있는지 확인
- isFull() : 큐가 풀로 차 있는지 확인
- peek() : 앞에 있는 원소를 삭제하지 않고 반환

# 큐 구현 및 활용

```
class Queue:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def enqueue(self, item):
        self.items.insert(_index: 0, item)

    def dequeue(self):
        return self.items.pop()

    def peek(self):
        return self.items[len(self.items)-1]

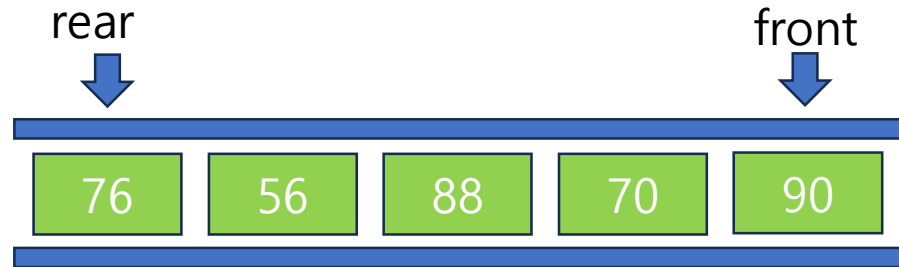
    def size(self):
        return len(self.items)
```

## ● 큐의 활용 예시

- 프린터 출력 처리
- 콜센터 대기시간
- 은행, 관공서 업무
- 프로세스 관리
- 캐시(Cache) 구현

## ● 큐의 문제점

- 요소를 제거하고 front와 rear의 위치를 이동해야 함
- 이동을 하지 않을 경우 해당 위치에 데이터가 없는 것을 계속 가르킴



■ 단순연결리스트로 큐 클래스를 완성하세요.  
큐 클래스를 활용한 큐 실행화면을 출력하세요.

- 과제 4월 29일(월) 까지 제출
- 제출방법 : [swcberet@naver.com](mailto:swcberet@naver.com)  
[메일제목] 학번\_성명\_데이터구조\_2차과제  
[파일명] 학번\_성명\_데이터구조\_2차과제
- 과제에 포함해야 할 내용  
전체 코드, 실제 동작결과 캡처