

큐

🕒 작성일시	@June 10, 2023 9:50 AM
📎 자료	
# 주차	3

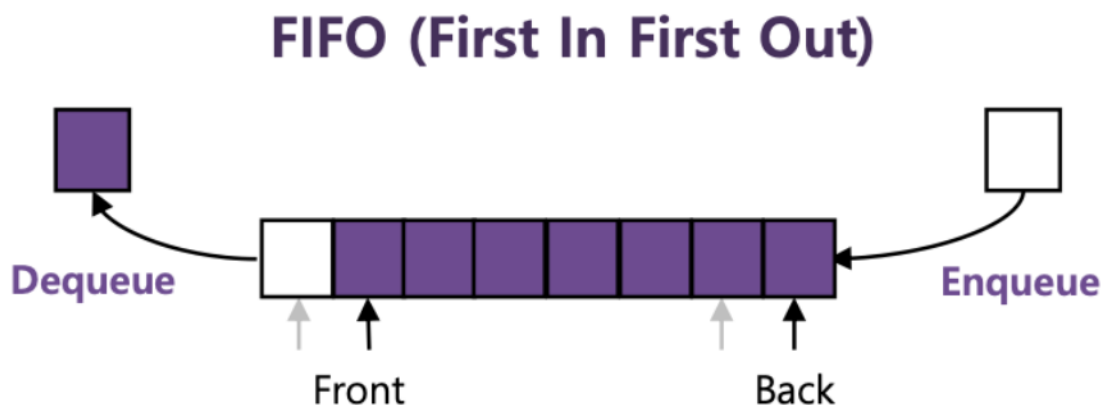
큐?

큐(Queue)는 **FIFO(First In First Out, 선입선출)** 원리에 따라 정렬된 컬렉션이다.

새원소는 뒤로 들어가서 앞에서 빠져나가는 구조다. 따라서 마지막에 추가된 원소는 큐의 뒷부분에서 제일 오래 기다려야 한다.

줄서기를 떠올리면 가장 쉽다.

컴퓨터에서는 '출력물 인쇄 대기'에 큐를 응용한다. 인쇄할 문서 5개를 각각 프로그램에서 열어 인쇄버튼을 누르면 5개의 문서가 차례대로 도착하는 데 프린터는 도착한 순서대로 문서를 열어 인쇄를 시작한다.



큐 구현

큐에서 사용하는 메서드

- enqueue(원소(들)): 큐의 뒤쪽에 원소(들)를 추가
- dequeue(): 큐의 첫번째 원소를 반환하고 큐에서 삭제
- front(): 큐의 첫 번째 원소를 반환하되 큐 자체는 그대로 둔다.
- isEmpty(): 큐가 비어있다면 true, 아니면 false반환
- size(): 큐의 원소 개수 반환

```
function Queue() {
  const items = [];

  this.enqueue = function(element) {
    items.push(element);
  }
  this.dequeue = function() {
    return items.shift();
  }
  this.front = function() {
    return items[0];
  }
  this.isEmpty = function() {
    return items.length === 0;
  }
  this.clear = function() {
    items = [];
  }
  this.size = function() {
    return items.length;
  }
}
```

우선순위 큐

우선순위 큐(priority queue)는 원소가 우선순위에 따라 추가되고 삭제되는 점이 다르다.

비슷한 경우를 실생활에서 생각해본다면 비행기 탑승을 떠올려보자. 1등석과 비즈니스석 승객은 항상 코치석 승객보다 우선순위가 높다. 어떤 국가에서는 연장자와 임산부에게 탑승 우선권을 주기도 한다.

병원 응급실도 그렇다 중상환자는 의산 판단에 따라 철과상을 입은 환자보다 먼저 진료실로 보낸다.

우선순위 큐는 두 가지 방법으로 구현할 수 있 있다.

1. 우선순위를 설정해 원소를 정확한 위치에 추가하는 것
2. 추가는 순서대로 하되 삭제만 우선순위에 따르는 것

```
// 우선순위 추가 큐
function priorityQueue() {
  const items = [];
  function QueueElement(element, priority) {
    this.element = element;
    this.priority = priority;
  }
  this.enqueue = function(element, priority) {
    const queueElement = new QueueElement(element, priority);
    if(items.length === 0) {
      items.push(queueElement);
    } else {
      const isAdded = false;
      for(let i=0; i<items.length; i++) {
        if(queueElement.priority < items[i].priority) {
          items.splice(i, 0, queueElement);
          isAdded = true;
          break;
        }
      }
      if(!isAdded) items.push(queueElement);
    }
  };
  // 그 밖의 메서드는 기본 큐의 구현체와 동일
}
```

위의 로직으로 구현한 우선 순위 큐를 **최소 우선순위 큐(min priority queue)**라고 부르는데, **우선 순위 값이 낮으면 낮을 수록 앞자리로 이동**하기 때문이다.

반대로 우선순위 값이 클수록 앞자리로 보내는 **최대 우선순위 큐(max priority queue)**도 있다.

환형큐(뜨거운 감자 게임)

뜨거운 감자게임이 환형 큐(circular queue)의 대표적인 예이다. 원을 그리고 서 있는 아이들이 뜨거운 감자를 옆사람에게 최대한 빨리 전달하다가 갑자기 모두 동작을 멈추고 그때 뜨거운 감자를 들고 있는 아이를 벌칙으로 퇴장시키는 게임이다. 최후의 1인이 남을 때까지 게임은 계속된다.

```
function hotPotato(nameList, num) {
  const queue = [];

  for(let i=0; i<nameList.length; i++) {
    queue.push(nameList[i]);
  }

  let eliminated = '';
  while(queue.length > 1) {
    for(let i=0; i<num; i++) {
      queue.push(queue.shift());
    }
    eliminated = queue.shift();
    console.log(`${eliminated} 를 뜨거운 감자 게임에서 퇴장시킵니다.`);
  }
  return queue.shift();
}
```