

큐

2021년 2월 19일 금요일 오후 7:14

252. 큐(queue)

• 큐(queue)

- 스택과 마찬가지로 자료의 삽입과 삭제에 대한 규칙이 있는 자료구조 중 하나
- 가장 먼저 자료구조에 삽입(enqueue)된 데이터가 제일 처음에 삭제(dequeue)됨



- 큐를 선입 선출(First In First Out)이라고도 함
- 스택과는 삭제 방향이 다름에 주목



• 큐를 현실에서 찾아본다면?

- 티켓 부스는 여러 개지만 한 줄 서기를 하는 경우



• 큐 역시 임의 접근은 안 됨!

- 언제나 켈 처음에 있는 자료만 제거 가능
- 중간 자료로 임의 접근 안 됨

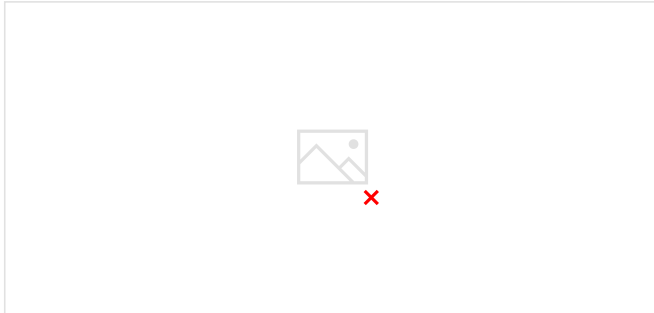


• 큐의 비효율적인 구현

- 그냥 배열을 사용하면 큐를 구현할 수는 있다
- enqueue하면 그냥 제일 뒤에 추가 $O(1)$



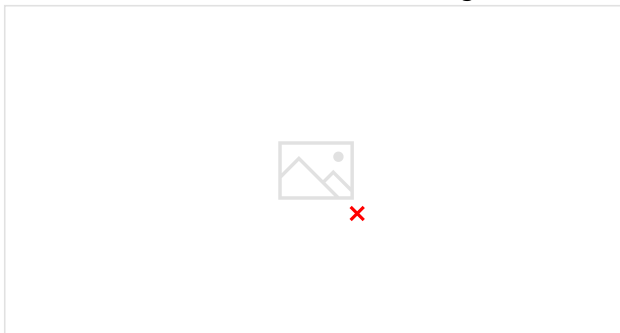
- dequeue하면 그냥 제일 앞에서 삭제 $O(n)$
 - 삭제하고 한칸씩 땡겨줘야 한다.. 그래서 비효율적



- 하지만, 앞에서 큐의 삭제도 $O(1)$ 이라 했는데?



- 큐의 삭제를 $O(1)$ 으로 만들려면
 - 내부적으로 배열을 사용하되 원형 버퍼(ring buffer)의 개념을 이용하면 가능



253. 큐의 삽입

- 큐의 삽입 예



- s_front
 - 어디서 빼와야 하나를 알려주는 변수
- s_back
 - 어디서 새롭게 줄을 세워야 하나 알려주는 변수
- 처음 코드 enqueue(10) 실행 시 변수 상태



- 마지막 코드 enqueue(80) 실행 시 변수 상태



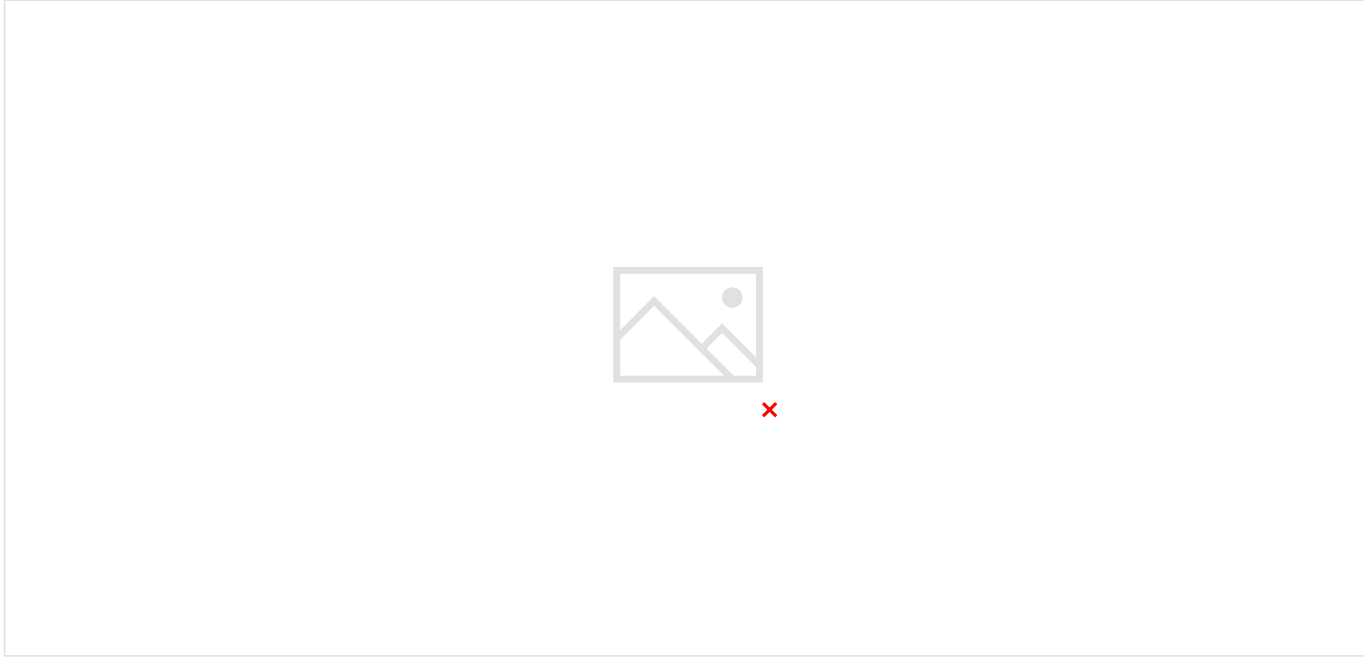
- s_num_count가 짝이었으므로 또 삽입하면 assert가 걸린다

• 큐의 삽입

- 보통 enqueue라고 표현
 - en: ~속에 넣다, queue: 큐
 - 큐에 넣다
 - 줄 젤 뒤에 세운다는 의미
 - 시간 복잡도: $O(1)$

254. 큐의 삭제

- 큐의 삭제 예



- 코드 실행 시 변수 상태



- 삭제하고 원소가 한칸 씩 이동하는 대신에 s_front가 증감하므로 $O(1)$

- 큐의 제거

- de: 분리/제거, queue: 큐
 - 큐에서 분리/제거하다
- dequeue라고 함
 - 앞에서 하나 빼온다고 해서
- 시간 복잡도: $O(1)$

255. 큐의 검색

- 큐의 검색

- 시간 복잡도: $O(n)$
- 젤 처음부터 찾을 때까지 뒤져야 함
- 스택과 마찬가지로 보통 enqueue()와 dequeue()만 허용하므로 임의의 요소에 접근할 방법이 없음

- 큐도 다 뽑아 확인하고 다시 원상복구 함

- 모든 요소를 다 제거했다가 다시 원상복구해야 함



- 중간까지만 제거하면 큐의 순서가 바뀔 따라서 전부 다 빼고 찾은다음 넣어야 한다



• 큐 검색의 시간 복잡도

- 스택과 똑같음
- 모든 요소를 다 제거했다가 다시 원상복구해야 함
- 그래서 제거에 $O(n)$ + 복구하는데 $O(n)$ 이 필요
- 따라서, $O(2n)$ 이지만 이건 그냥 $O(n)$ 라고 말함

• 큐의 용도

- 현실 세계에서 대기줄이 필요한 경우에 다 적용 가능
- 데이터 유입 속도가 데이터 소모 속도보다 빠른 경우
- 데이터 제공자의 수가 데이터를 소비자의 수와 다른 경우
 - 예: 은행 창구는 여럿인데 줄은 한 줄로만 설 때
 - 멀티 쓰레딩에서도 이런 일들을 함
- 입출력 스트림 버퍼링도 같은 개념