

스택, 큐, 덱의 이해

송기태 (kitae040522@gmail.com)
Soongsil Univ. (Computer Science and Engineering)

Content

스택 (Stack)

큐 (Queue)

덱 (Deque)



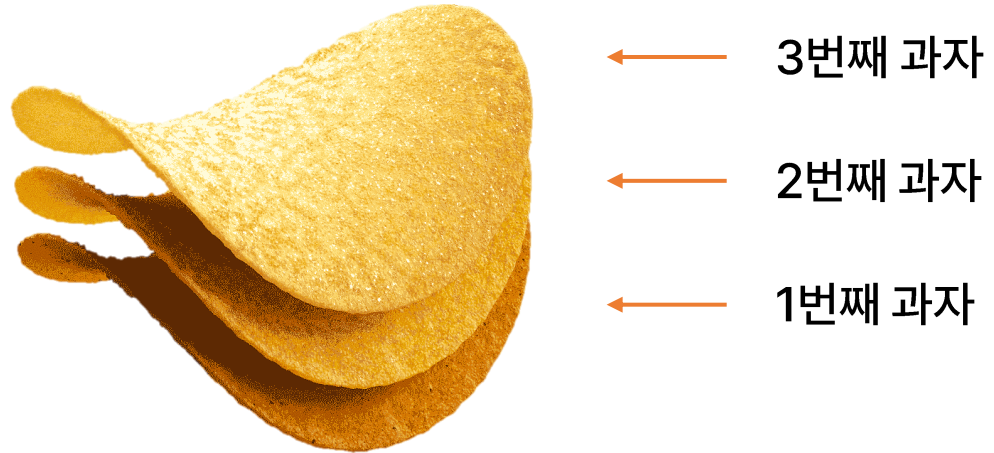
스택은 프링글스와 비슷하다.

스택 (Stack)



처음 넣은 과자를 먹으려면 가장 마지막에 먹을 수 있다

스택 (Stack)



처음 넣은 과자를 먹으려면 가장 마지막에 먹을 수 있다

이걸 영어로
LIFO (Last In First Out) 구조라고 부른다

스택도 마찬가지로 LIFO 구조이다

스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

얘는 대충 프링글스 통으로 생각하자



스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

데이터 #1

프링글스 통에 '데이터 #1'을 저장하려면, 입구에서 하나 떨어트리면 된다.



스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

데이터 #1

프링글스 통에 '데이터 #1'을 저장하려면, 입구에서 하나 떨어트리면 된다.

데이터 #1

스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

데이터 #2

'데이터 #2'을 저장하려면, 똑같이 입구에서 하나 떨어트리면 된다.

데이터 #1

스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

데이터 #2

'데이터 #2'을 저장하려면, 똑같이 입구에서 하나 떨어트리면 된다.

데이터 #2

데이터 #1

스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

그 후....

새로운 데이터를 넣을 공간이 없다면, 위에서부터 하나씩 빼주면 된다.

데이터 #8
데이터 #7
데이터 #6
데이터 #5
데이터 #4
데이터 #3
데이터 #2
데이터 #1

스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

데이터 #8

'데이터 #8'을 삭제하고, 새로운 데이터를 넣는다.

데이터 #7
데이터 #6
데이터 #5
데이터 #4
데이터 #3
데이터 #2
데이터 #1

스택 (Stack)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

데이터 #8

'데이터 #8'을 삭제하고, 새로운 데이터를 넣는다.

데이터 #9

데이터 #7

데이터 #6

데이터 #5

데이터 #4

데이터 #3

데이터 #2

데이터 #1

데이터를 프링글스 통에 넣는 연산을 **PUSH**라고하며,
데이터를 빼는 연산을 **POP**이라고 한다

스택의 컨셉을 이해했는가?
다음은 큐를 이해해보도록 하자.

FIVE GUYS®

큐는 음식점에 들어가기 위해 줄 스는 것이라고 생각하면 된다

큐 (Queue)



먼저 줄 서 있던 사람이 음식점에 먼저 들어갈 수 있다

큐 (Queue)



마지막에 줄 서 있던 사람은 음식점에 마지막으로 들어간다

이걸 영어로
FIFO (First In First Out) 구조라고 부른다

큐도 마찬가지로 FIFO 구조이다

큐 (Queue)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

큐 (Queue)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



얘는 대충 대기열로 생각하자

큐 (Queue)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



대기열에 '데이터 #1'을 저장하려면, 대기열에 아무도 없으니 출구와 가장 가까운 곳에 저장하면 된다.

큐 (Queue)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



대기열에 '데이터 #2'을 저장하려면, 대기열에 '데이터 #1'이 있으므로 그 뒤에 저장하면 된다.

큐 (Queue)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



그 후.... 새로운 데이터를 넣을 공간이 없다면, 출구에서 하나씩 빼주면 된다.

큐 (Queue)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



그 후.... 새로운 데이터를 넣을 공간이 없다면, 출구에서 하나씩 빼주면 된다.

'데이터 #1'을 출구로 내보내고, '데이터 #10'을 입구로 넣어준다.

데이터를 대기열에 넣는 연산을 **ENQUEUE**라고하며,
데이터를 빼는 연산을 **DEQUEUE**이라고 한다

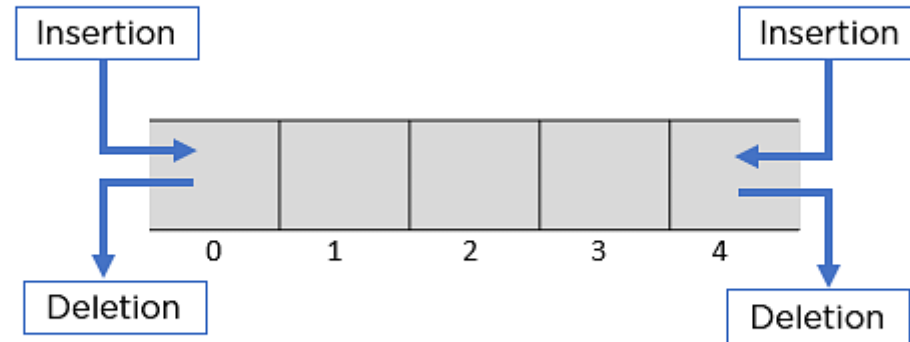
큐의 컨셉을 이해했는가?
다음은덱을 이해해보도록 하자.

덱 (Deque)



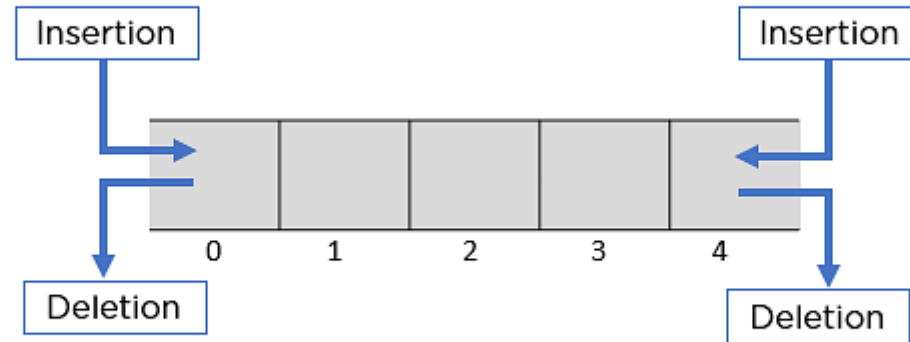
덱은 큐와 스택을 짬뽕 시킨 거로 생각하면 된다.

덱 (Deque)



입구와 출구가 각각 두 개씩 존재한다.

덱 (Deque)



양방향 큐라고 생각하면 될 것이다.

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



↑
얘는 대충 대기열로 생각하자

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



대기열에 '데이터 #1'을 저장하려면, 왼쪽 입구나 오른쪽 입구에 저장하면 된다.

지금은 오른쪽 입구에 데이터를 저장했다.

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



대기열에 '데이터 #1'을 저장하려면, 왼쪽 입구나 오른쪽 입구에 저장하면 된다.

지금은 왼쪽 입구에 데이터를 저장했다.

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



대기열에 '데이터 #2'을 저장하려면, 대기열에 '데이터 #1'이 있으므로 그 뒤에 저장하면 된다.

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



그 후.... 새로운 데이터를 넣을 공간이 없다면, 양 옆에서 원하는 만큼 빼주면 된다.

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



그 후.... 새로운 데이터를 넣을 공간이 없다면, 양 옆에서 원하는 만큼 빼주면 된다.

지금은 '데이터 #1'을 오른쪽 출구로 내보내고, '데이터 #10'을 왼쪽 입구로 넣어준 상황이다.

덱 (Deque)

이해를 돕기 위해, 그림으로 설명하도록 하겠다.



그 후.... 새로운 데이터를 넣을 공간이 없다면, 양 옆에서 원하는 만큼 빼주면 된다.

지금은 '데이터 #10'을 왼쪽 출구로 내보내고, '데이터 #11'을 오른쪽 입구로 넣어준 상황이다.

데이터를 대기열 왼쪽에 넣는 연산을 **ENQUEUE_FRONT**라고하며,
데이터를 왼쪽에서 빼는 연산을 **DEQUEUE_FRONT**이라고 한다

데이터를 대기열 오른쪽에 넣는 연산을 **ENQUEUE_REAR**라고하며,
데이터를 왼쪽에서 빼는 연산을 **DEQUEUE_REAR**이라고 한다

덱 자료구조가 실생활에서 어떻게 활용될 수 있을까?
실생활 예시를 생각해보는 것이 오늘의 과제

Thank You!

송기태 (kitae040522@gmail.com)
Soongsil Univ. (Computer Science and Engineering)