**스택**

텍스트, 스크린샷, 도표, 디자인이(가) 표시된 사진

자동 생성된 설명

●스택이란?

-주어진 일련의 데이터들을 층층이 쌓아 올리는 형태의 자료구조

-스택의 최상단에 위치하는 top에서 원소의 삽입 및 삭제가 이루어짐

-Last In First Out – 마지막으로 들어온 데이터가 먼저 나감

Ex)쌓여있는 책 더미

●스택의 operations

-push(): 스택의 top에 데이터를 새롭게 추가함

-pop(): 스택의 top에 위치하는 데이터를 제거한 후 값을 반환함

-peek(): 스택 top에 존재하는 데이터를 제거하지 않고 값을 반환함

-isFull(): 스택이 다 찼는지 확인

-isEmpty(): 스택이 비어 있는지 확인

●스택과 재귀함수

-재귀(recursion)함수: 함수의 정의부에서 자기 자신을 호출하는 함수

-프로그램 내에서 함수 혹은 메서드가 호출되는 방식은 스택 자료구조 차원에서 접근할 수 있음. 함수가 최초로 호출되었을 때 그 함수는 스택에 push되는 걸로 간주함. 함수의 실행이 종결되었을 때 그 함수는 스택에서 pop되는 걸로 간주함. 현재 돌아가고 있는 함수가 스택에 최상단에 위치함.

●링크드 리스트를 이용한 스택 구현(파이썬)

텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명

●배열을 이용한 스택 구현(파이썬)

-스택 생성

stack = list() #비어있는 리스트를 생성함으로써 빈 스택에서 출발

-push()

stack.append(3) #append를 이용하여 리스트의 맨 끝에 원소를 삽입

-pop()

stack.pop #pop을 이용하여 가장 마지막에 삽입된 원소를 제거, 삭제된 원소 반환

-peek()

stack[-1]

●스택의 활용

-괄호 판별

-중위 표현식 -> 후위 표현식

**큐**

**텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명**

-First In First Out – 먼저 들어온 데이터가 먼저 나감

-queue는 대기열이라는 뜻으로, 큐의 작동 방식은 대기열과 유사함

●큐의 operations

-enqueue(): 큐의 rear에 새로운 데이터를 삽입

-dequeue(): 큐의 front에 존재하는 데이터를 제거

-peek(): 큐의 front에 있는 데이터를 제거하지 않고 값을 반환함

-isFull(): 큐가 다 찼는지 확인

-isEmpty(): 큐가 비어 있는지 확인

●링크드 리스트를 이용한 큐 구현(파이썬)

텍스트, 문서, 메뉴, 스크린샷이(가) 표시된 사진

자동 생성된 설명텍스트, 문서, 메뉴, 스크린샷이(가) 표시된 사진

자동 생성된 설명

●배열을 이용한 큐 구현(파이썬)

-큐 생성

queue = list() #비어있는 리스트를 생성함으로써 빈 큐에서 출발

-enqueue()

queue.append(7) #append를 이용하여 리스트의 맨 끝에 원소를 삽입

-dequeue()

del queue[0] #최전방 원소는 최좌측에 있으므로 가장 첫 원소를 제거

-peek()

queue[0] -> 순서상 맨 끝 원소가 아닌, 가장 첫 원소임에 주목

●큐의 문제점(배열로 구현할 경우)

-데이터를 삭제할 경우, 뒤쪽 데이터들을 모두 한 칸씩 밀어야만 원하는 삭제가 가능하므로, 데이터 삭제에 대한 시간복잡도는 O(n)

●환형 큐(Circular Queue)

-선형이 아닌 환형 대기열의 형태로 데이터를 제어하는 자료구조

-삭제에 비교적 높은 시간 복잡도를 요하는 일반 큐 대비 우월

-일반적으로 크기를 고정해둔 상태에서 환형 큐를 이용

원, 스크린샷, 데이터 저장 장치, 다채로움이(가) 표시된 사진

자동 생성된 설명원, 텍스트, 콤팩트 디스크, 데이터 저장 장치이(가) 표시된 사진

자동 생성된 설명

-front: 주어진 환형 큐의 최전방 원소와 함께 움직임(dequeue할 때 움직임)

-rear: 주어진 환형 큐의 최후방 원소와 함께 움직임(enqueue할 때 움직임)

-환형 큐에서 enqueue를 시도할 경우 rear가 가리키고 있었던 위치에 데이터가 삽입되며 새 데이터 삽입으로 인해 rear은 한 칸 밀려나게 됨 -> O(1)

콤팩트 디스크, 원, 텍스트, 다채로움이(가) 표시된 사진

자동 생성된 설명

-환형 큐에서 deque를 시도할 경우 front가 가리키고 있던 위치의 데이터가 삭제되며(None) 데이터 삭제로 인해 front는 한 칸 밀려나게 됨 -> O(1)