

## 자료구조 활용

### - 가장 큰 수(스택)

- 스택: LIFO
- 입력 받은 숫자를 앞에서부터 하나씩 읽을 때 앞자리 숫자가 현재 읽은 숫자보다 작으면 그 숫자를 지우고 현재 읽은 숫자가 그 자리를 대체한다.
- 현재 읽은 숫자보다 크거나 같다면 현재 읽은 숫자를 앞자리 숫자 뒤에 붙인다.(append)
- 모든 연산이 끝나고 지우는 횟수가 남아있다면 리스트의 맨 뒤에서부터 남은 횟수까지를 slice하여 출력한다.(stack[: -m], m은 남은 횟수)
- 지우는 횟수를 다 사용하면 나머지 숫자를 다 읽는다.
- **스택 사용** - 파이썬에서는 리스트와 리스트의 pop을 이용하여 스택을 사용한다.

### - 쇠막대기(스택)

- 닫는 괄호를 만나면 바로 이전 괄호가 여는 괄호인지 확인하여 레이저인지 판단한다.

### - 후위표기식 만들기(스택)

- 중위표기식을 후위표기식으로 변경

### - 후위식 연산(스택)

- 후위표기식 계산

### - 공주 구하기(큐)

- 큐: 먼저 들어온 것이 먼저 나감 - FIFO
- deque를 이용하여 큐 구현
  - ◆ **append** - 리스트 맨 뒤에 삽입, **pop** - 리스트 맨 뒤에서 삭제, **appendleft** - 리스트 맨 앞에 삽입, **popleft** - 리스트 맨 앞에 삭제

## - 응급실(큐)

- any() 함수: 인자 중 하나라도 참이면 True 리턴!
- 환자의 위험도 뿐만 아니라 순서도 고려해야하므로 데이터와 데이터의 index를 하나로 묶어서 표현하는 것이 좋다. → 튜플 사용

## - 교육과정 설계(큐)

- 데이터뿐만 아니라 순서도 고려해야한다.
- 수업 계획과 교육 과정을 나타내는 각각의 큐가 필요하다.
- for if break else문 활용

## - 단어 찾기(해쉬)

- 해쉬: 딕셔너리 or 리스트 사용 → key에 대응하는 value가 존재

## - 아나그램(딕셔너리 해쉬), 아나그램(리스트 해쉬)

## - 최소힙

- 완전이진트리로 구현
- 부모 노드의 값이 자식 노드의 값보다 작게 트리를 구성
- 루트 노드는 트리에서 값이 가장 작음
- 트리가 비어 있다면 그냥 입력
- 트리가 비어 있지 않다면 입력 숫자와 부모 노드의 숫자를 비교하여 입력
- 힙에서 pop하면 루트 노드를 삭제하고 그 자리에 트리의 마지막 level에서 마지막 숫자를 부모 노드로 가져와 최소힙을 적용한다.
- 힙에서 push하면 트리의 마지막 level에 삽입하고 부모 노드와 비교하여 최소힙을 만든다.
- import heapq as hq → 기본적으로 최소힙으로 동작함.

## - 최대힙

- import heapq as hq → 입력 받는 숫자에 -1을 곱하여 push하면 최대힙으로 동작함