

Chapter 01.

기본 자료구조

핵심 유형 문제풀이

핵심 유형 문제풀이 | 다양한 문제를 접하며 코딩 테스트에 익숙해지기
강사 나동빈

Chapter 01. 기본 자료구조

핵심 유형 문제풀이

Ch1. 기본 자료구조 혼자 힘으로 풀어보기

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

문제 제목: 스택

문제 난이도: 하(Easy)

문제 유형: 자료구조, 스택, 시뮬레이션

추천 풀이 시간: 20분

Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

- 본 문제는 스택(stack) 자료구조를 구현할 수 있다면 쉽게 풀 수 있는 문제입니다.
- Python에서는 단순히 리스트(list) 자료구조를 사용하여 스택을 구현할 수 있습니다.
- 원소를 삽입할 때는 append() 메서드를 사용합니다.
- 원소를 삭제할 때는 pop() 메서드를 사용합니다.

Ch1. 기본 자료구조 소스 코드

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

```
import sys
input = sys.stdin.readline # 빠른 입력 함수 사용

n = int(input()) # 명령의 수 입력
stack = [] # 스택(stack) 초기화

for _ in range(n):
    command = input().strip().split(' ') # 공백 기준 구분
    if command[0] == 'push': # 원소 삽입
        stack.append(int(command[1]))
    elif command[0] == 'pop': # 원소 삭제
        if len(stack) == 0: print(-1) # 원소가 없는 경우
        else: print(stack.pop())
    elif command[0] == 'size': # 원소의 개수 출력
        print(len(stack))
    elif command[0] == 'empty': # 스택이 비어있는지 출력
        if len(stack) == 0: print(1) # 원소가 없는 경우
        else: print(0)
    elif command[0] == 'top': # 최상단 원소 출력
        if len(stack) == 0: print(-1) # 원소가 없는 경우
        else: print(stack[-1])
```

Ch1. 기본 자료구조 혼자 힘으로 풀어보기

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

문제 제목: 제로

문제 난이도: 하(Easy)

문제 유형: 자료구조, 스택, 시뮬레이션

추천 풀이 시간: 20분

Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

- 본 문제는 스택(stack) 자료구조를 구현할 수 있다면 쉽게 풀 수 있는 문제입니다.
- 입력으로 0이 들어오는 경우 가장 최근에 입력된 원소를 제거합니다.
- 이는 스택(stack)의 pop() 함수와 동일한 기능으로 이해할 수 있습니다.

Ch1. 기본 자료구조 소스 코드

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

```
import sys
input = sys.stdin.readline # 빠른 입력 함수 사용

k = int(input()) # 수의 개수
stack = [] # 스택(stack) 초기화

for _ in range(k):
    x = int(input()) # 하나씩 수 입력
    if x == 0: # 값이 0인 경우 가장 최근 수 제거
        stack.pop()
    else: # 스택에 해당 수 삽입
        stack.append(x)

print(sum(stack))
```


Ch1. 기본 자료구조 혼자 힘으로 풀어보기

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

문제 제목: 스택 수열

문제 난이도: 하(Easy)

문제 유형: 자료구조, 스택, 시뮬레이션

추천 풀이 시간: 30분

Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

- 본 문제는 스택(stack) 자료구조를 활용해 해결할 수 있는 문제입니다.
- 다양한 예시를 생각해 보며 시뮬레이션 해보면 아이디어를 떠올리기 수월합니다.

Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

- 주어진 수열: [4, 3, 6, 8, 7, 5, 2, 1]
- ① 먼저 4를 꺼내야 하므로, 1, 2, 3, 4를 차례대로 스택에 넣고 4를 꺼냅니다.
현재 스택: [1, 2, 3] / 넣은 원소: [1, 2, 3, 4]
- ② 이어서 3을 꺼냅니다.
현재 스택: [1, 2] / 넣은 원소: [1, 2, 3, 4]
- ③ 이어서 6을 꺼내야 하므로, 5, 6을 스택에 넣고 6을 꺼냅니다.
현재 스택: [1, 2, 5] / 넣은 원소: [1, 2, 3, 4, 5, 6]

Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

- 주어진 수열: [4, 3, 6, 8, 7, 5, 2, 1]

④ 이어서 8을 꺼내야 하므로, 7, 8을 스택에 넣고 8을 꺼냅니다.

현재 스택: [1, 2, 5, 7] / 넣은 원소: [1, 2, 3, 4, 5, 6, 7, 8]

[아이디어 캐치]

이어서 7을 꺼내는 건 가능할까요? **가능**합니다.

하지만 1, 2, 5를 꺼내는 것은 **불가능**합니다.

다시 말해 스택의 최상위 원소 top()보다 작은 값을 꺼내는 것은 불가능합니다.

Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

- 본 문제는 스택(stack)을 이용하여 시뮬레이션 하듯이 풀 수 있는 문제입니다.
 1. 기본적으로 특정한 수 x 가 될 때까지 스택에 넣고, 이후에 x 를 스택에서 뺍니다.
 2. 만약 스택의 $\text{top}()$ 보다 x 가 더 작다면, 해당 수열을 만드는 것이 **불가능**합니다.

Ch1. 기본 자료구조 소스 코드

핵심 유형 문제풀이

Ch1.

핵심 유형 문제풀이

```
import sys
# 빠른 입력 함수 사용
input = sys.stdin.readline

n = int(input()) # 수의 개수
stack = [] # 스택(stack) 초기화
answer = [] # 최종 정답
current = 1 # 현재 삽입할 수
```

```
for _ in range(n):
    x = int(input()) # 하나씩 수 입력
    # top()보다 x가 더 크다면, 스택에 삽입
    while len(stack) == 0 or stack[-1] < x:
        stack.append(current)
        current += 1
        answer.append('+')
    # top()과 x가 같다면, 스택에서 제거
    if stack[-1] == x:
        stack.pop()
        answer.append('-')
    # top()보다 x가 더 작다면, 불가능
    else:
        answer = ['NO']
        break

for x in answer: # 정답 출력
    print(x)
```