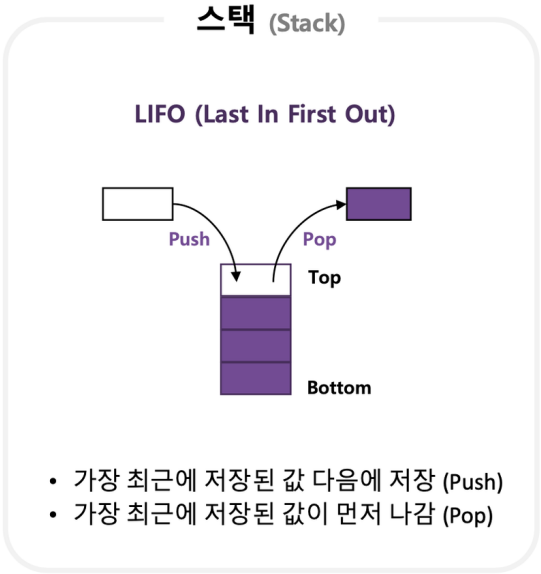
이영직

- STACK
  - 백준_10828
    - #1차 풀이
    - 고려사항
    - #2차풀이
  - 백준_9012
    - #1차 풀이
  - 프로그래머스_기능개발
    - #1차 풀이
    - #2차풀이
  - 프로그래머스_프로세스
    - #1차풀이
    - 고려사항
    - 다른 방안



STACK

백준_10828

정수를 저장하는 스택을 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 다섯 가지이다.

- push X: 정수 X를 스택에 넣는 연산이다.
- pop: 스택에서 가장 위에 있는 정수를 빼고, 그 수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 스택에 들어있는 정수의 개수를 출력한다.
- empty: 스택이 비어있으면 1, 아니면 0을 출력한다.
- top: 스택의 가장 위에 있는 정수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.

#1차 풀이

```
n = int(input())
stack=[]

while n != 0:
  a= input().split()
  if a[0] == 'push':
    stack.append(a[1])
    n -=1
  elif a[0] == 'top':
    if len(stack) ==0:
```

```
      print(-1)
      n -=1
    else:
      print(stack[-1])
      n -=1
  elif a[0] == 'size':
    print(len(stack))
    n -=1
  elif a[0] == 'empty':
    if len(stack) ==0:
      print(1)
      n -=1
    else:
      print(0)
      n -=1
  elif a[0] == 'pop':
    if len(stack) ==0:
      print(-1)
      n -=1
    else:
      print(stack.pop())
      n -=1
```

시간초과 오류로 인한 실패

고려사항

- 입력 시간 줄이기
  - 입출력 속도 비교 : sys.stdin.readline > raw_input() > input( )
- while 문 대신 for문 사용해보기
- 파이썬은 스택 구조를 List 형태로 풀어야함

#2차풀이

```
import sys
n = int(sys.stdin.readline())
stack=[]
for i in range(n):
  a= sys.stdin.readline().split()
  if a[0] == 'push':
    stack.append(a[1])

  elif a[0] == 'top':
    if len(stack) ==0:
      print(-1)
    else:
      print(stack[-1])

  elif a[0] == 'size':
    print(len(stack))

  elif a[0] == 'empty':
    if len(stack) ==0:
      print(1)
```

```
    else:
      print(0)

  elif a[0] == 'pop':
    if len(stack) ==0:
      print(-1)
    else:
      print(stack.pop())
```


백준_9012

#1차 풀이

괄호 문자열(Parenthesis String, PS)은 두 개의 괄호 기호인 '(' 와 ')' 만으로 구성되어 있는 문자열이다. 그 중에서 괄호의 모양이 바르게 구성된 문자열을 올바른 괄호 문자열(Valid PS, VPS)이라고 부른다. 한 쌍의 괄호 기호로 된 "( )" 문자열은 기본 VPS 이라고 부른다. 만일 x 가 VPS 라면 이것을 하나의 괄호에 넣은 새로운 문자열 "(x)"도 VPS 가 된다. 그리고 두 VPS x 와 y를 접합(concatenation)시킨 새로운 문자열 xy도 VPS 가 된다. 예를 들어 "(())()"와 "((()))" 는 VPS 이지만 "(()(", "(())())" , 그리고 "(()" 는 모두 VPS 가 아닌 문자열이다.

여러분은 입력으로 주어진 괄호 문자열이 VPS 인지 아닌지를 판단해서 그 결과를 YES 와 NO 로 나타내어야 한다.

```
#백준9012
n =int(input())

for i in range(n):
  a = input()
  stack = 0
  for j in a:
    if j == '(':
      stack +=1
    else:
      stack -=1
      if stack < 0:
        break
  if stack==0:
    print('Yes')
  else:
    print('No')

```

프로그래머스_기능개발

#1차 풀이

```
def solution(progresses, speeds):
    answer = []
    anw=[]
    import math
    for i in range(len(progresses)):
        anw2 = math.ceil((100-progresses[i] )/ speeds[i])
        anw.append(anw2)
    print(anw)

```

```
#     b=0
#     count =0
#     while len(anw) !=0:
#         if anw[0] > b:
#             if count > 0:
#                 answer.append(count)
#                 count = 0
#             elif anw[0] == b:

#                 count+=1
#                 anw.pop(0)
#             else:
#                 b=anw.pop(0)
#                 count +=1
#         elif anw[0] < b:
#             count+=1
#             anw.pop(0)
#     answer.append(count)


```

#2차풀이

```
def solution(progresses, speeds):
    answer = []
    anw=[]
    import math
    for i in range(len(progresses)):
        anw2 = math.ceil((100-progresses[i] )/ speeds[i])
        anw.append(anw2)
    print(anw)

    b= anw[0];
    count = 0;
    for a in anw:
        if a <= temp:
            count+=1
        else:
            answer.append(count)
            count = 1
            b = a

    answer.append(count)
    return answer
```

프로그래머스_프로세스

#1차풀이

deque에 대한 아이디어가 떠오르지 않아 풀이를 보고 해결 했음

```
def solution(priorities, location):
    answer = 0
    from collections import deque

    d = deque([(v,i) for i,v in enumerate(priorities)])
```

```
6
7       while len(d):
8           item = d.popleft()
9           if d and max(d)[0] > item[0]:    #pop 이후 d의 원소가 비어 있으면 안됨
10              d.append(item)
11          else:
12              answer += 1
13              if item[1] == location:
14                  break
15      return answer
```

고려사항

- deque의 사용 방법 - from collections import deque
- `enumerate` 로 인덱스와 한번에 묶어주기

다른 방안

- any사용하기
  - 한번도 보지 못한 방법 알아두면 아주 유용할 듯

```
1  # def solution(priorities, location):
2  #     queue =  [(i,p) for i,p in enumerate(priorities)]
3  #     answer = 0
4  #     while True:
5  #         cur = queue.pop(0)
6  #         if any(cur[1] < q[1] for q in queue):
7  #             queue.append(cur)
8  #         else:
9  #             answer += 1
10 #             if cur[0] == location:
11 #                 return answer
12
```