스택

list = [ ]

Push : append

Pop : pop

size : len

empty : len

top : index -1

제로

```
stack = [   ]

if input == 0 :
        stack.pop()
else:
        stack.append(input)

print [ sum(stack))
```

괄호

```
standard = 0
loop
  if "(":
      standard +
  if ")":
      standard -
  if standard < 0:
      print(NO)
      break
if standard == 0:
    print(YES)
elif standard > 0:
      print(NO)
```

균형잡힌 세상

[, (⇒ push , flag = true

],) ⇒ stack[-1]과 일치 ⇒ pop

불일치 ⇒ flag false

empty stack ⇒ flag false

flag true & empty stack

⇒ yes

스택 수열

input = [1,2,5,3,4]

stack: [3,4

1,2,5

m = 0

for i in range(input):

if input[i] not in stack:

m += 1

stack.append(m)

result.append(+)

result = [ ]

elif input[i] in stack & stack[-1] == input[i]:

stack.pop()

result.append(-)

else:

print(NO)

break.

오큰수

input [3 5 2 7] ans [5 7 7 -1] input [9 5 4 8]
ans [-1 8 8 -1]

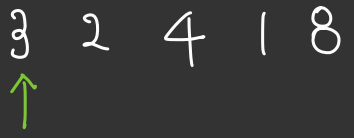
NGE(i) :

```
for i in range( len(input)-1):
    if input[i] > max(input[i+1:]):
        ans.append(-1)
        continue

    for num in input[i+1:]:
        if input[i] < num:
            ans.append(num)
            break

ans.append(-1)
print(" ".join(ans))
```

시간 초과 풀이
시간복잡도 = $O(n^2)$

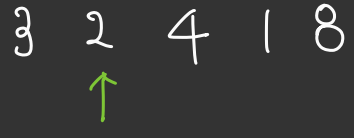
$O(n^2) \rightarrow O(n)$ 스택 활용

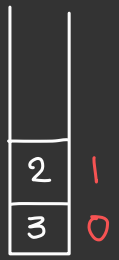
stack = [숫자, 인덱스]



오큰수

-1	-1	-1	-1	-1





오큰수

-1	-1	-1	-1	-1

오큰수

4	4	-1	-1	-1

3 2 4 1 8

오큰수

4	4	-1	-1	-1

3 2 4 1 8

오큰수

4	4	8	8	-1