

<스택>

1. 문제

문제: A의 오른쪽에 있으면서 A보다 큰 수 중 가장 왼쪽에 있는 수

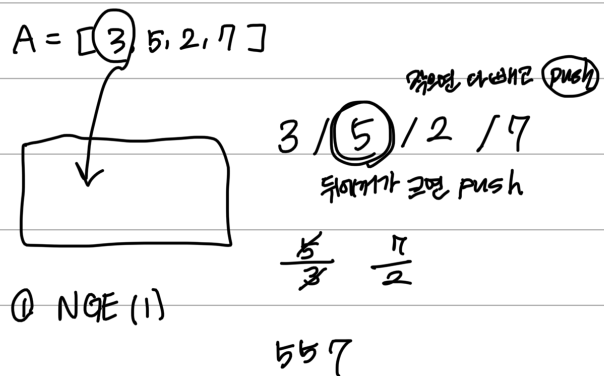
$A = [3, 5, 2, 7]$

$NGE(1) = 5$ $NGE(3) = 7$

$NGE(2) = 7$ $NGE(4) = -1$

Try 1 실패 (시간초과)

```
1 import sys
2 n = int(sys.stdin.readline())
3 num_list = list(map(int, sys.stdin.readline().split()))
4 max = max(num_list)
5 result = []
6
7 for i, v in enumerate(num_list):
8     if v == max or v == num_list[-1]:
9         result.append(-1)
10    else:
11        for j in num_list[i:]:
12            if j > v:
13                result.append(j)
14                break
15
16 for i in result:
17     print(i, end = " ")
```



<큐, 덱>

* 큐 (queue) : 표준 사다리꼴로 들어선 사람들의 줄

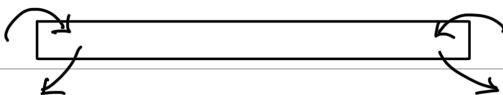
→ 선형선형의 자료구조 (FIFO)



push : 자료 넣기 front : 가장 앞의 자료 empty : 비어있는지 확인

pop : 자료 빼기 back : 가장 뒤의 자료 size : 자료 개수

* 덱 : 양쪽 끝에서 삽입과 삭제가 모두 가능한 자료구조



3. 카드 2

1 1
2 2 2
4 2 3 2 2

함수형이 버리고

짝수형이로 아래로

6 2 3 4 2 4
3 3 4 2 4 4

8 2 3 4 5 2 4 2
3 3 4 5 5 4 4 2
4 4 5 5 2 4 4 2

10 2 3 4 5 6 2 4 6 4
3 3 4 5 6 2 4 6 4
4 4 5 6 2 4 6 4
5 5 6 2 4 6 4 4

③ 1 3 → ②

④ 1 3 2 → ④

⑤ 1 3 5 4 → ②

⑥ 1 3 5 2 6 → ④

⑦ 1 3 5 7 4 2 → ⑥

⑧ 1 3 5 7 2 6 4 → ⑧

⑨ 1 3 5 7 9 4 8 2 → 6

12 2 3 4 5 6 7 2 4 6 2 6 6
3 3 4 5 6 7 2 4 6 2 6 6
4 4 5 6 7 2 4 6 2 6 6 6
5 5 6 7 2 4 6 2 6 2 6 6
6 6 7 2 4 6 2 6 2 6 6 6
7 7 2 4 6 2 6 2 6 6 6 6

7 8 2 4 6
8 2 4 6
① 2 4 8
4 6 8
② 6 4 8
8 4 8
8

1 2 3
2 3 4
3 4 5
4 5 6
5 6 7
6 7 8
7 8 2

9 2 4 6 8
2 4 6 8
③ 2 4 6 8
2 4 6 8
4 6 8 2
6 8 2 6
8 2 6 2 6

3	2	9	2	15	14
4	4	10	4	16	16
5	2	11	6	17	2
6	4	12	8	18	
7	6	13	10	19	
8	8	14	12	20	

2	4	8	16	32	64	128	...
↓	↓	↓	↓	18			
2	4	8	16	18			
			$18-16=2$	20			
			$2 \times 2 = 4$	$20-16=4$			
				$2 \times 4 = 8$			

3. 요세푸스 문제

(1,1)	1	
(2,1)	1, 2	
(2,2)	2, 1	1 2 3 4 5 6 7
(3,1)	1, 2, 3	1 2 3
(3,2)	2, 1, 3	1 2 3
(3,3)	3, 1, 2	
(4,1)	1, 2, 3, 4	
(4,2)	2, 4, 3, 1	
(4,3)	3, 2, 4, 1	
(4,4)	4, 1, 3, 2	