

알튜비튜

스택, 큐, 덱

오늘은 STL에서 제공하는 container adaptor인 stack과 queue 그리고 sequence container인 deque에 대해 알아봅니다.
가장 대표적이면서 가장 중요하기도 한 자료구조들 입니다.

/<> 17298번 : 오큰수 - Gold 4

문제

- 크기가 N 인 수열 $A = A_1, A_2, \dots, A_N$ 에서 수열의 각 원소 A_i 에 대해 오큰수 (해당 수의 오른쪽에 있으면서 해당 수보다 큰 수 중에서 가장 왼쪽에 있는 수를 의미) 구하기

제한 사항

- 조건에 해당하는 수가 없을 시 오큰수는 -1가 됨

예제 입력1

```
4
3 5 2 7
```

예제 출력1

```
5 7 7 -1
```

문제 팁

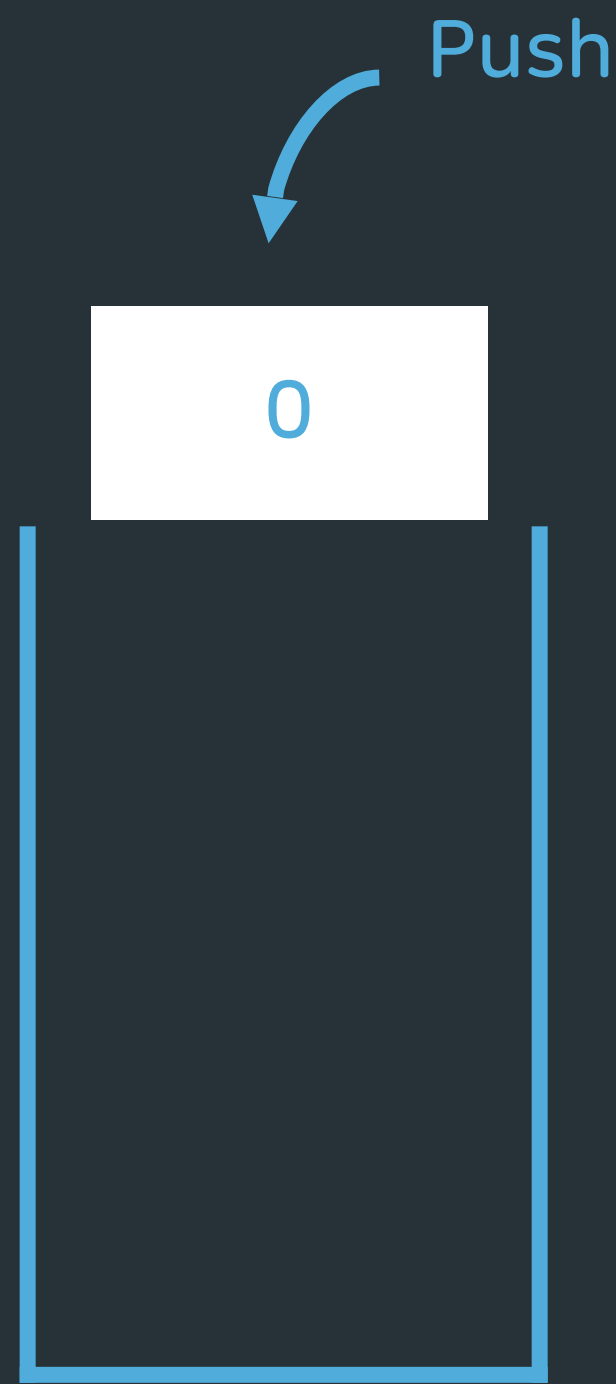
- 스택에 새로 들어오는 수가 top에 존재하는 수보다 크면 그 수는 오큰수가 됨
- 오큰수를 구한 후 수열에서 오큰수가 존재하지 않는 숫자 (스택에 남은 수들)의 오큰수는 -1임
- 스택에는 수열의 값이 아닌 인덱스를 저장

문제 해결

1. 수열A의 데이터값이 스택의 top 데이터값보다 크면 해당 스택 값을 pop 하고 해당 수의 오큰수에 수열A의 데이터값 저장
2. 위 과정을 조건을 만족하는 한 스택의 수들에 대해 진행
3. 현재 수열A의 인덱스를 스택에 push
4. 위 과정을 수열 길이만큼 반복한 후 스택에 남은 인덱스의 오큰수에 -1 저장

수열 A

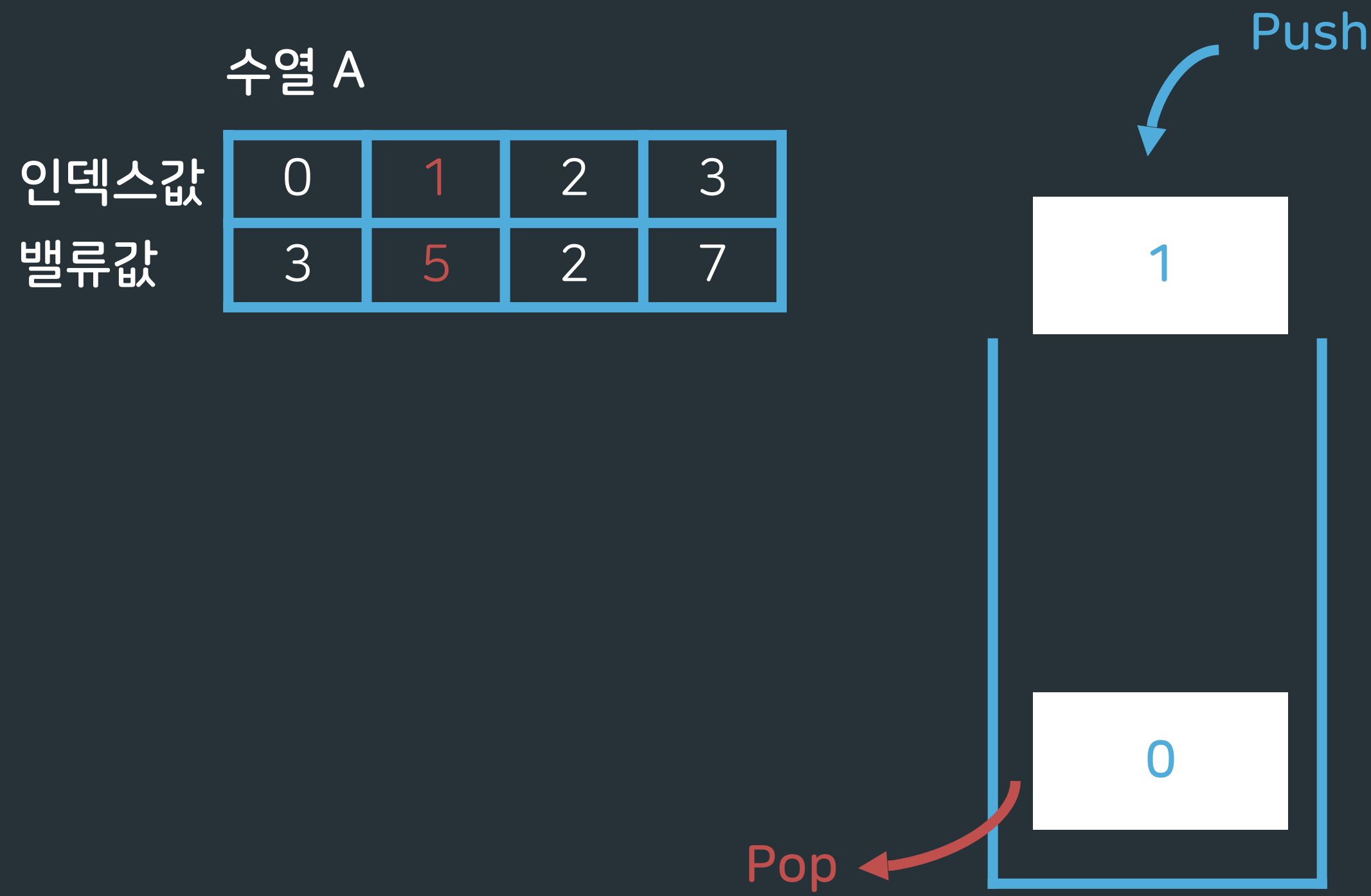
인덱스값	0	1	2	3
밸류값	3	5	2	7



결과값

인덱스값	0	1	2	3
밸류값				

예제

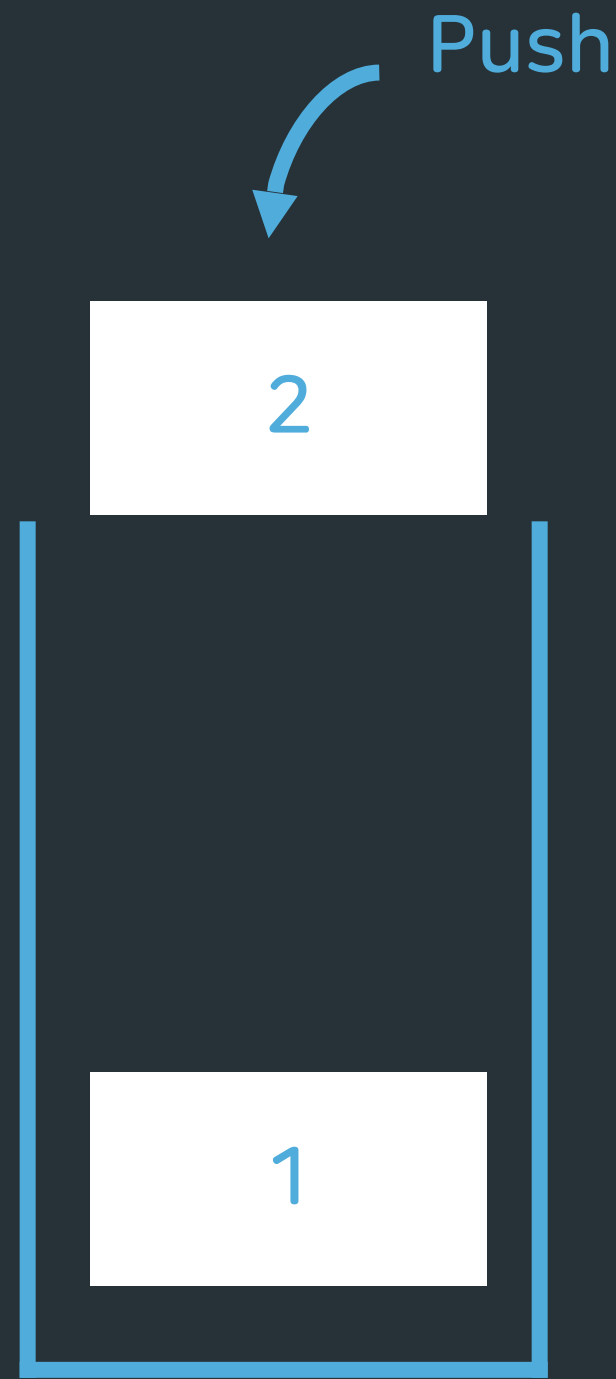


결과값

인덱스값	0	1	2	3
밸류값	5			

수열 A

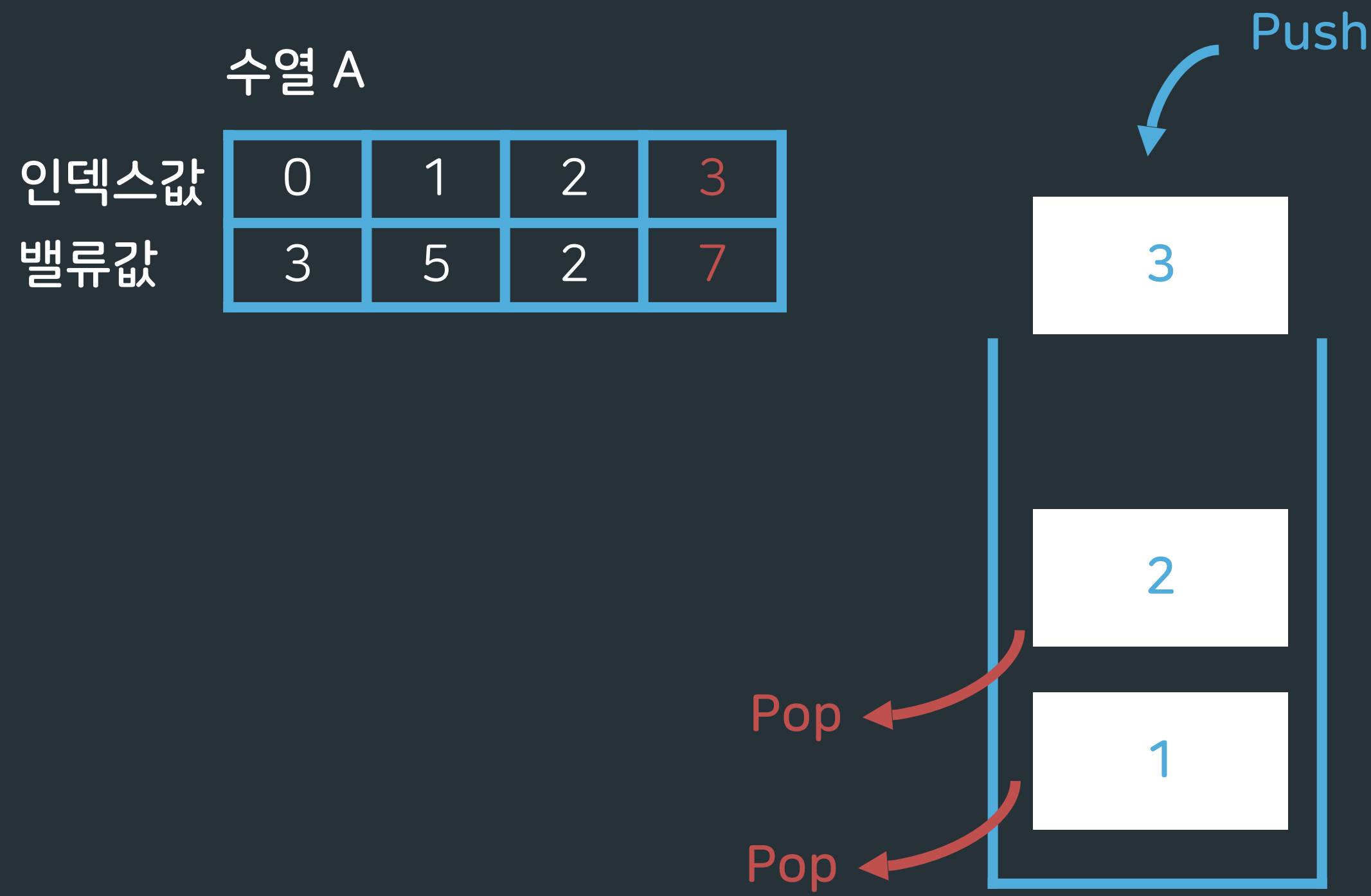
인덱스값	0	1	2	3
밸류값	3	5	2	7



결과값

인덱스값	0	1	2	3
밸류값	5			

예제



결과값

인덱스값	0	1	2	3
밸류값	5	7	7	

수열 A

인덱스값	0	1	2	3
밸류값	3	5	2	7



결과값

인덱스값	0	1	2	3
밸류값	5	7	7	-1

/<> 1918번 : 후위 표기식 - Gold 2

문제

- 중위 표기식이 주어졌을 때 후위 표기식으로 고치는 프로그램 만들기

제한 사항

- 수식의 피연산자는 알파벳 대문자
- -A와 같이 연산자가 피연산자 앞에 오거나 AB와 같이 * 가 생략되는 수식은 주어지지 않음
- 표기식은 알파벳 대문자와 +, -, *, /, (,)로만 이루어짐
- 길이는 100을 넘지 않음

예제 입력1

A*(B+C)

예제 출력1

ABC+*

중위 표기법

- 연산자를 두 연산자 사이에 표기하는 방법으로 가장 일반적으로 사용됨
- ex. $A+B$

전위 표기법

- 연산자를 먼저 표시하고 연산자에 필요한 피연산자를 나중에 표기하는 방법
- ex. $+AB$

후위 표기법

- 피연산자를 먼저 표시하고 연산자를 나중에 표기하는 방법
- Ex. $AB+$

중위 표기식 → 후기 표기식

$A + B * C$



연산자 우선순위에 따라
괄호로 묶기

$(A + (B * C))$



$(A + (B * C))$

연산자를
괄호의 오른쪽으로 옮겨
주기



$A B C * +$

$A + B * C$



$A B C * +$

Hint

1. 피연산자의 순서는 변하지 않아요!
2. 연산자 우선순위에 따라 연산자의 순서를 변화시켜 볼까요?

연산자(+, -, *, /, (,)) 우선순위

$$(A + (B * C)) \rightarrow A B C * +$$

곱셈(*), 나눗셈(/)을 덧셈(+)과 뺄셈(-)보다 먼저 수행한다

→ 곱셈과 나눗셈이 덧셈과 뺄셈보다 우선순위가 높다!



*, / 의 우선순위 > +, - 의 우선순위

중위표기식

$A*(B+C)$

$A*(B+C)$

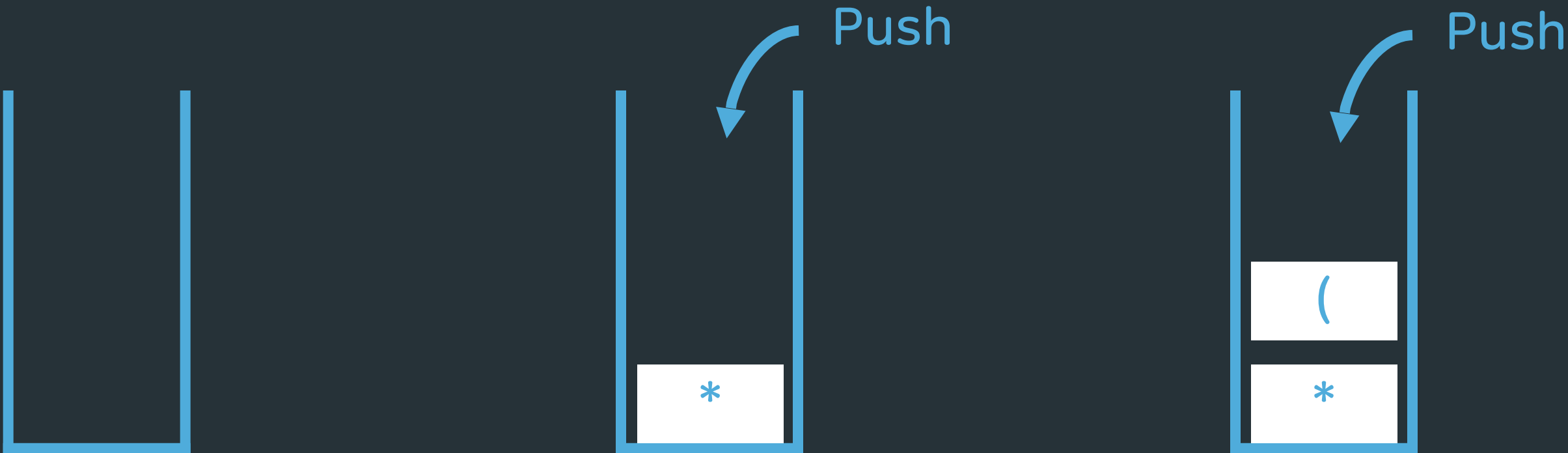
$A*(B+C)$

후위표기식

A

A

A

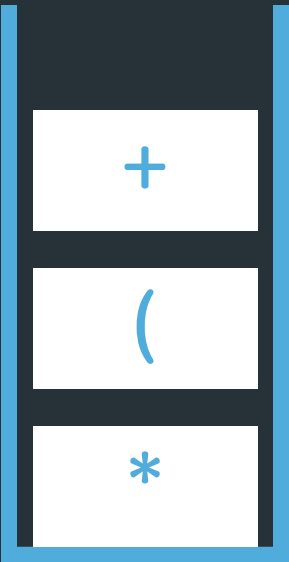
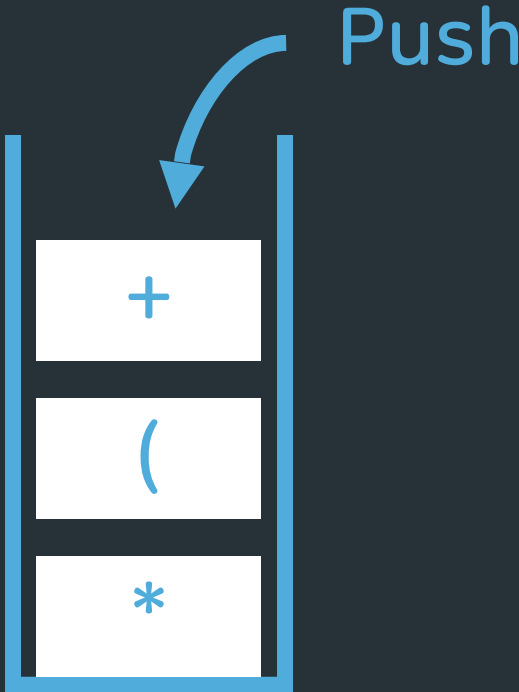
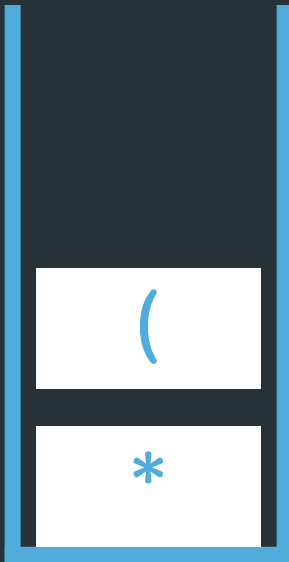


중위표기식

$A*(B+C)$

$A*(B+C)$

$A*(B+C)$



후위표기식

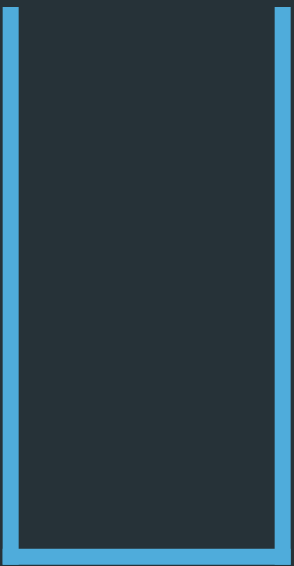
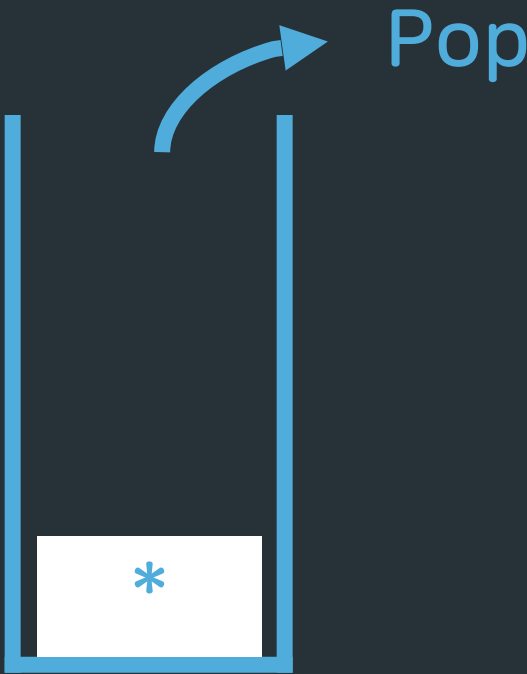
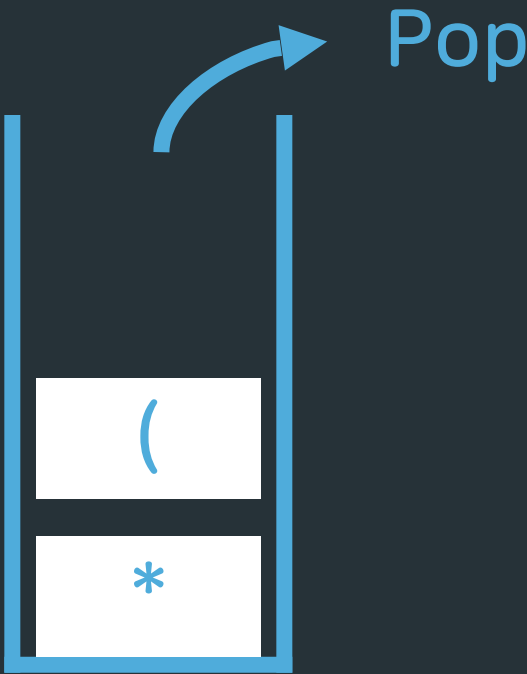
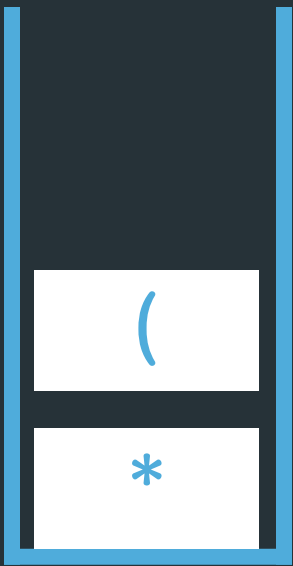
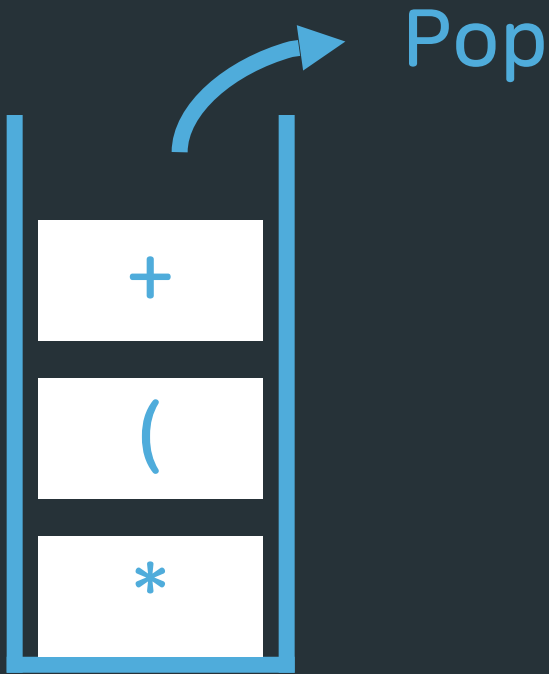
AB

AB

ABC

중위표기식

$A*(B+C)$



후위표기식

ABC

ABC+

ABC+

ABC+

ABC+*

중위표기식

$A+B*C-D/E$

$A+B*C-D/E$

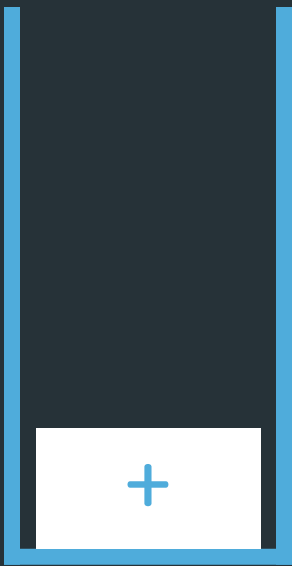
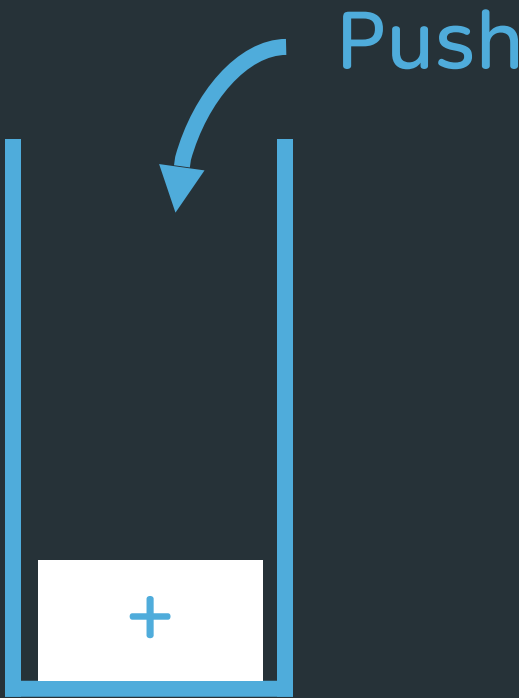
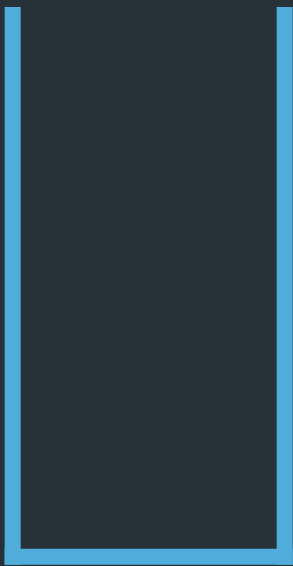
$A+B*C-D/E$

후위표기식

A

A

AB

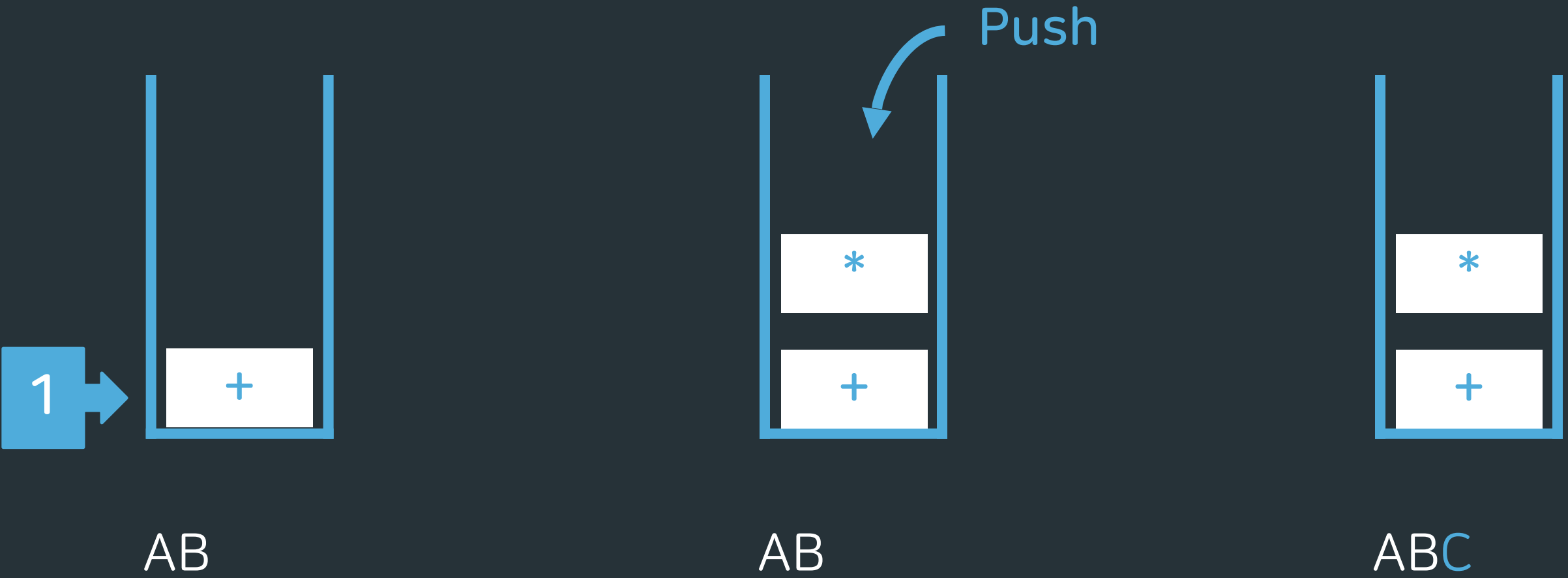


중위표기식

2
↓
 $A+B * C - D / E$

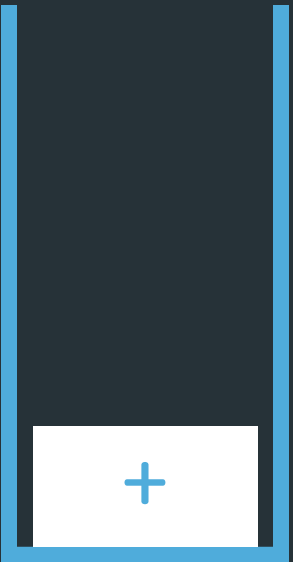
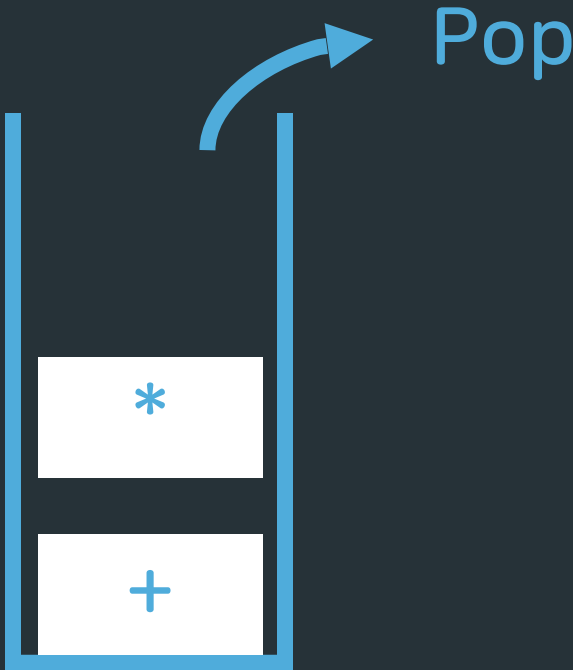
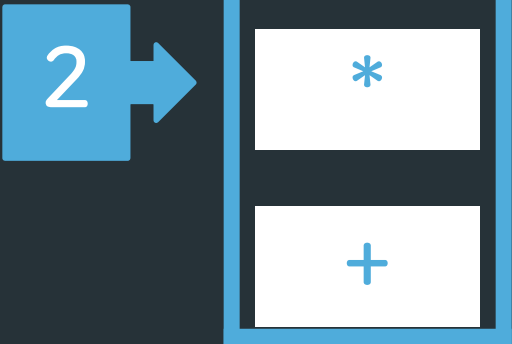
$A+B * C - D / E$

후위표기식



중위표기식

1
↓
 $A+B*C-D/E$



후위표기식

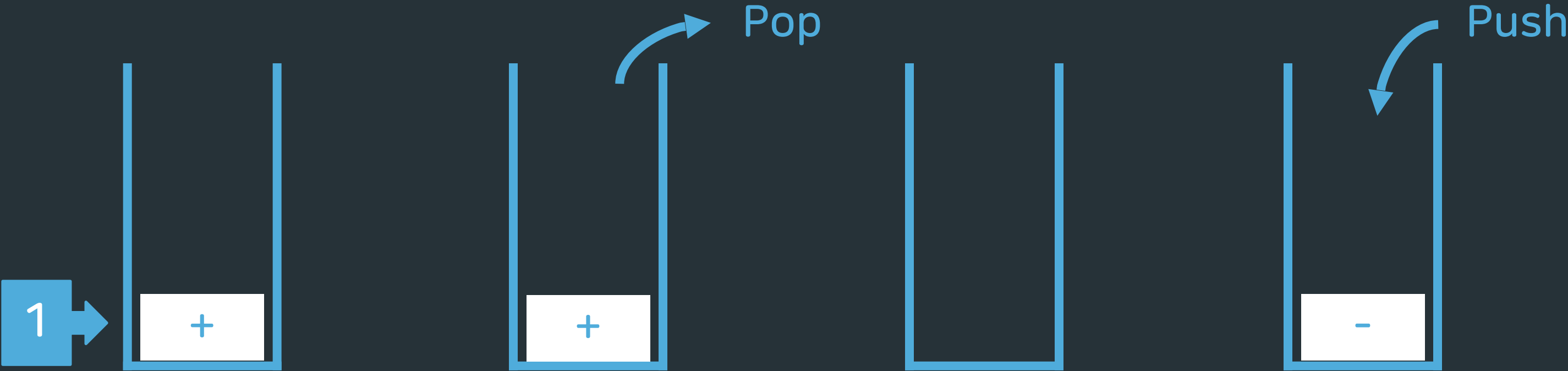
ABC

ABC

ABC*

중위표기식

1
↓
 $A+B*C-D/E$



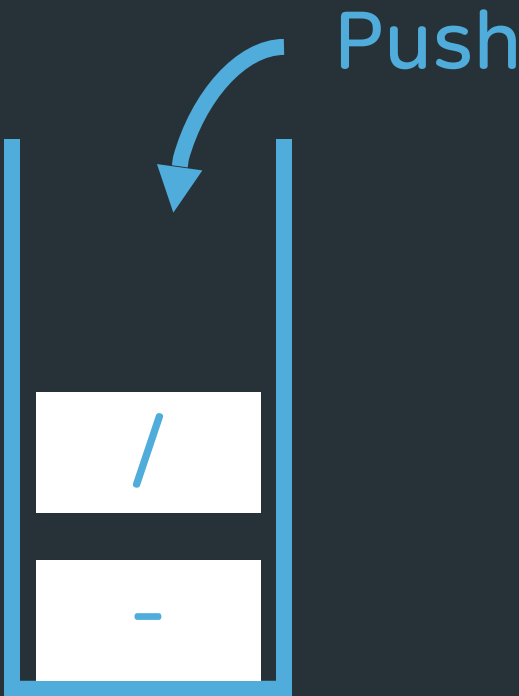
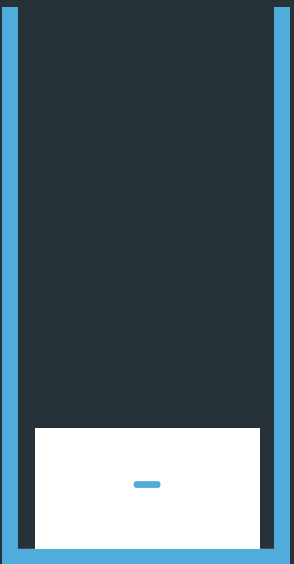
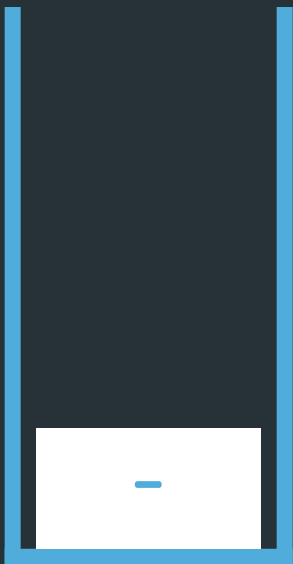
후위표기식

ABC* ABC* ABC*+ ABC*+

중위표기식

$A+B*C-D/E$

$A+B*C-D/E$



후위표기식

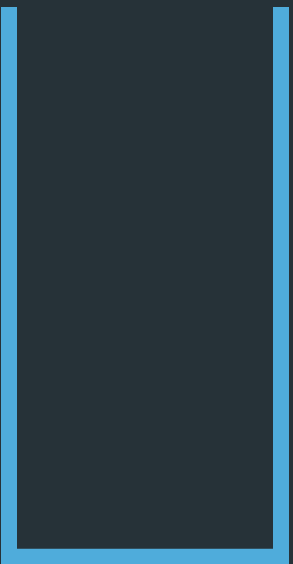
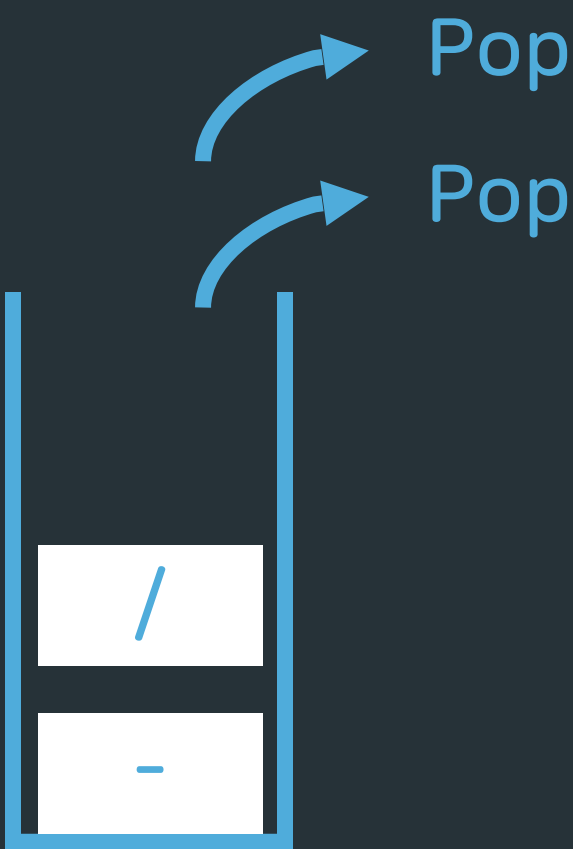
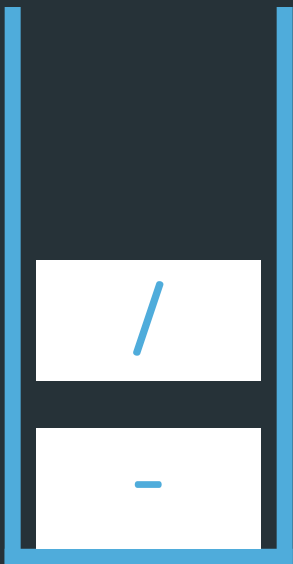
$ABC*+D$

$ABC*+D$

$ABC*+D$

중위표기식

$A+B*C-D/E$



후위표기식

$ABC*+DE$

$ABC*+DE$

$ABC*+DE/-$

/<> 10757번 : 큰 수 A + B - Bronze 5

문제

- 두 정수 A와 B를 입력받은 다음, A+B를 출력하는 프로그램을 작성하시오.

제한 사항

- A와 B의 입력 범위는 $0 < A, B < 10^{10000}$

예제 입력1

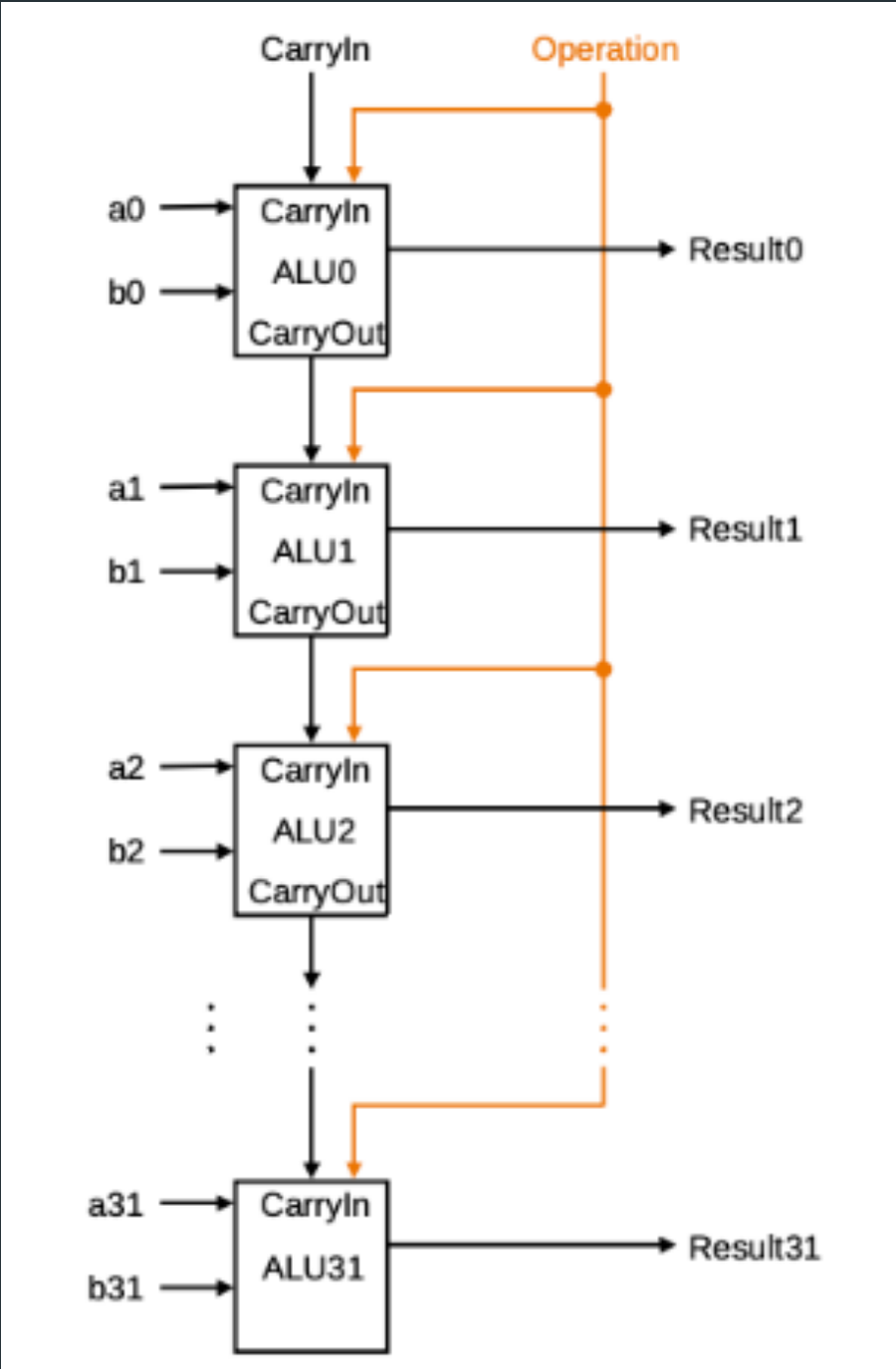
```
9223372036854775807 9223372036854775808
```

예제 출력1

```
18446744073709551615
```

- A,B가 최대 10^{10000}
- int : 최대 $2^{31} - 1$, long : 최대 $2^{63} - 1$
- $\text{int} < \text{long} \lllllllllll 10^{10000}$
- 따라서, 덧셈을 직접 구현하자

덧셈의 원리?!



$$\begin{array}{r} A = 54321 \\ B = 65432 \end{array}$$

$$\begin{array}{r} A = 456321 \\ B = \quad 5432 \end{array}$$

$$\begin{array}{r} A = 1234 \\ B = 65432 \end{array}$$

따라서 $A + B$ 를 할 때,

A 의 한자리 수 + B 의 한자리 수 + carry(올림수)

⇒ 연산을 모든 자리 수에 대해서 해주면 된다!

⇒ $O(N)$ 의 시간복잡도

수행 시간 계산해봅시다!

어떤 수 X 의 자리 수를 아는 방법은 무엇인가요?

1 ~ 9 : 1자리

10 ~ 99 : 2자리

.

.

$10^{(N-1)} \sim 10^N - 1$: N 자리

X 는 최대 10^{10000} 이므로 10001 자리

$O(N) \rightarrow 10001$ 번의 계산 < 1초

추가로 풀어보면 좋은 문제!

/<> 2164번 : 카드2 - Silver 4

/<> 18115번 : 카드 놓기 - Silver 3

/<> 4889번 : 안정적인 문자열 - Silver 1

/<> 17299번 : 오등큰수 - Gold 3

+ 백준 "단계별 풀이" 에서 '스택 큐 덱' 파트 문제들