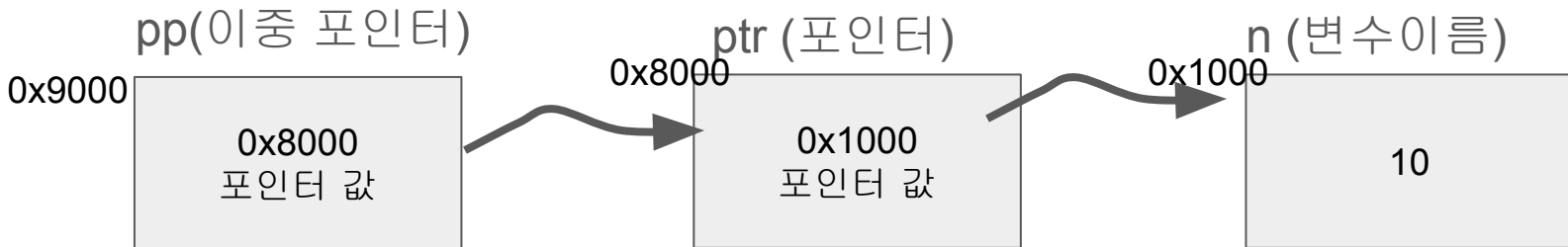
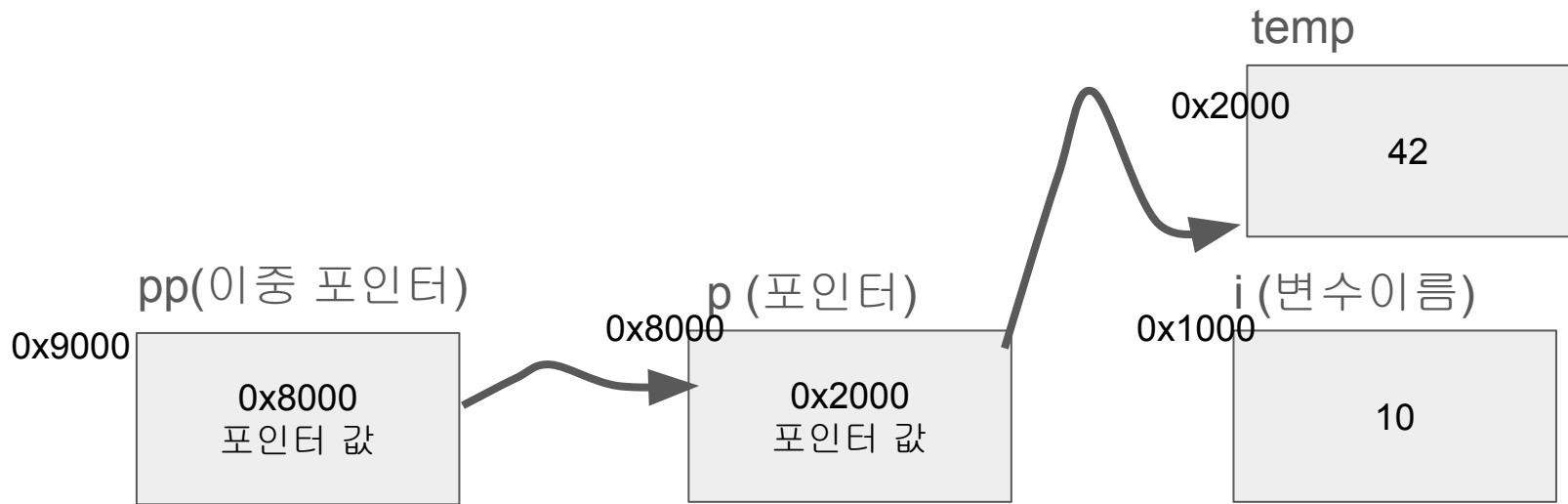


# C Algorithm

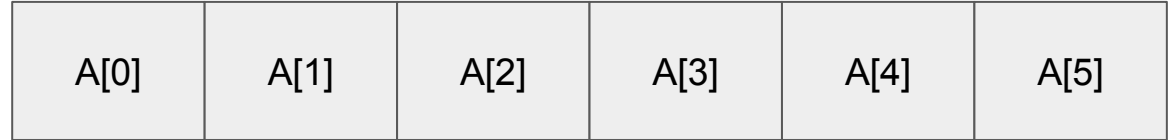
```
int n = 10;  
int* ptr = &n;  
int** pp = &ptr;
```





int A[6]

A



int A[2][3]

A

	0열	1열	2열
0행	A[0][0]	A[0][1]	A[0][2]
1행	A[1][0]	A[1][1]	A[1][2]

```
char st[16] = "Hello World";
```

'H'	'e'	'l'	'l'	'o'		'W'	'o'	'r'	'l'	'd'	'\0'				
-----	-----	-----	-----	-----	--	-----	-----	-----	-----	-----	------	--	--	--	--

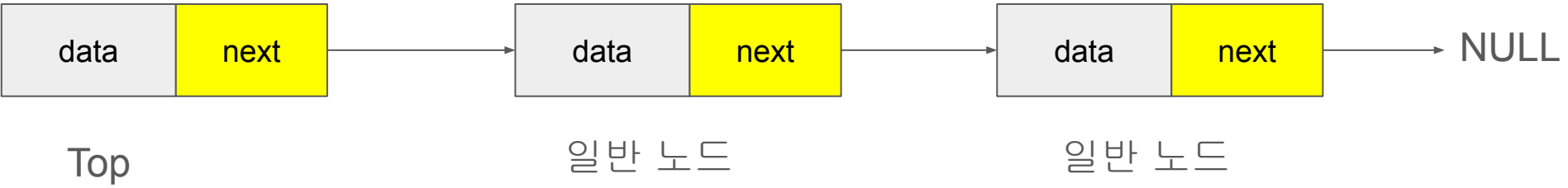
[0] [1]

[15]

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()



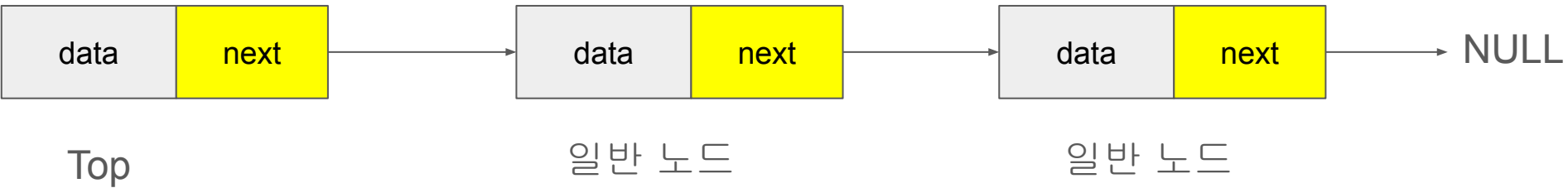
스택 - 링크드 리스트 구현  
=> 스택 삽입 과정1





# 스택 - 링크드 리스트 구현

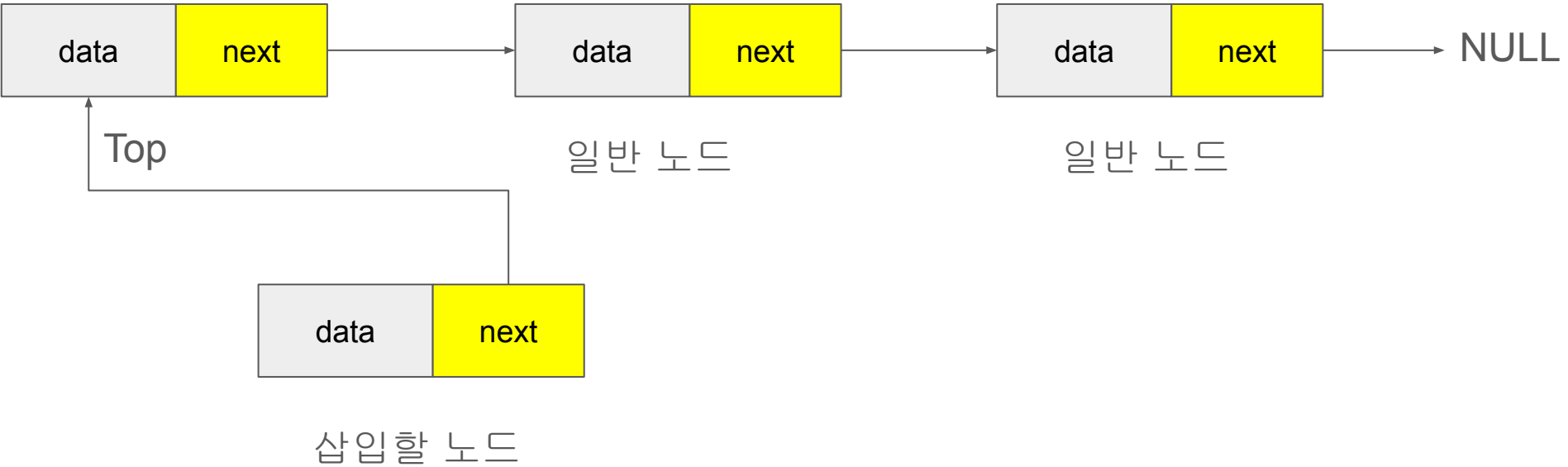
## => 스택 삽입 과정2



삽입할 노드

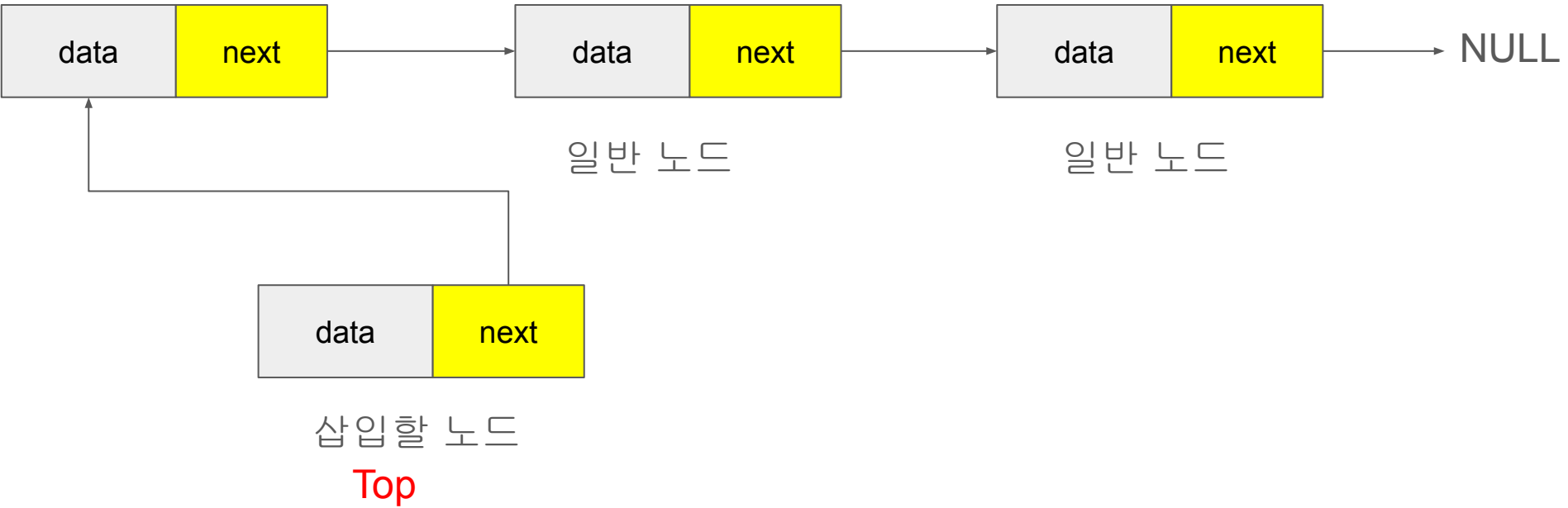
# 스택 - 링크드 리스트 구현

## => 스택 삽입 과정3



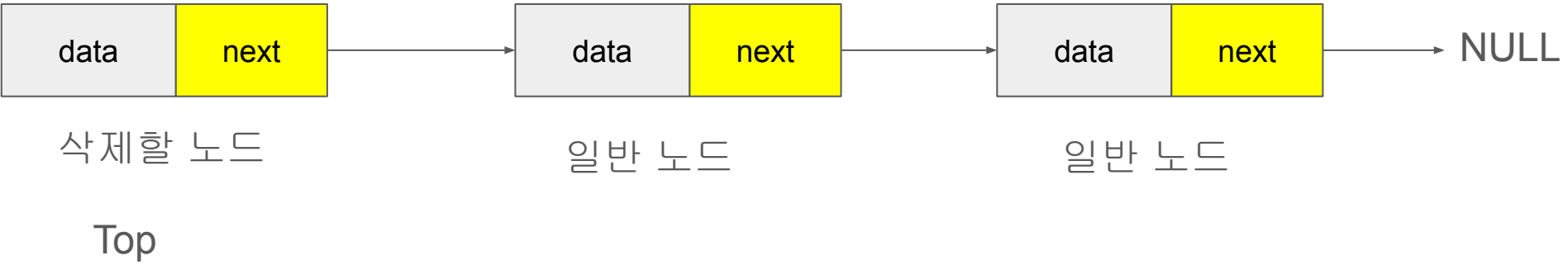
# 스택 - 링크드 리스트 구현

## => 스택 삽입 과정4



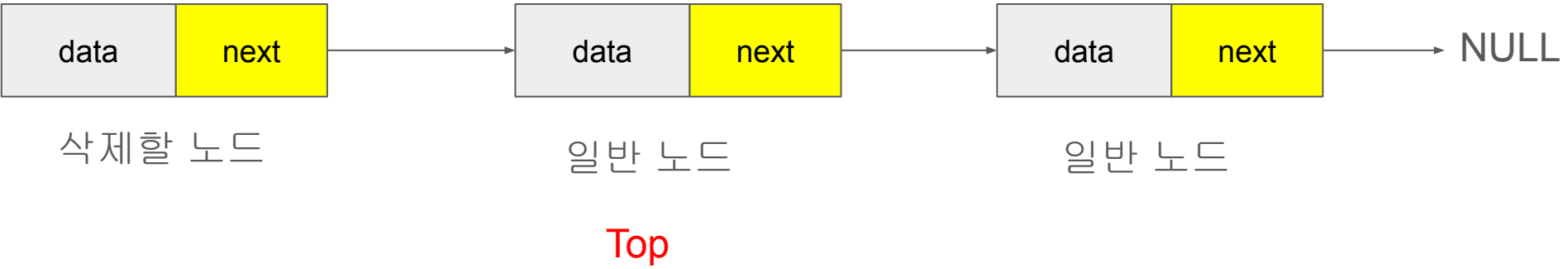
# 스택 - 링크드 리스트 구현

=> 스택 추출 과정1

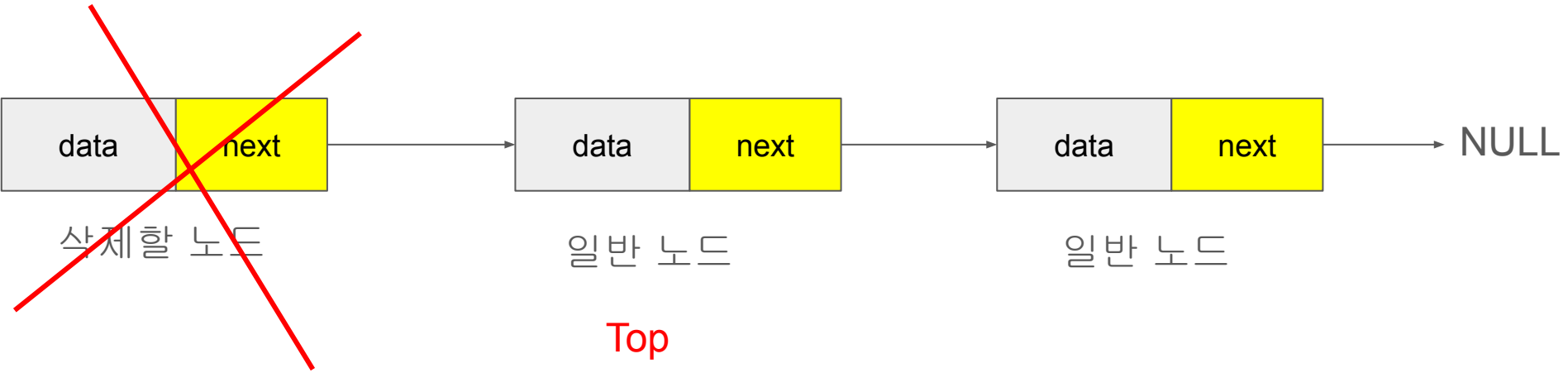


# 스택 - 링크드 리스트 구현

## => 스택 추출 과정2



스택 - 링크드 리스트 구현  
=> 스택 추출 과정3



# 큐(Queue)

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()

출구(앞)



입구(뒤)

# 큐(Queue)

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()

출구(앞)



입구(뒤)



# 큐(Queue)

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()

출구(앞)



입구(뒤)

# 큐(Queue)

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()

출구(앞)



입구(뒤)

# 큐(Queue)

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()

출구(앞)



입구(뒤)

## 큐(Queue)

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()

출구(앞)



입구(뒤)

# 큐(Queue)

PUSH(7) PUSH(5) PUSH(4) POP() PUSH(6) POP()

출구(앞)

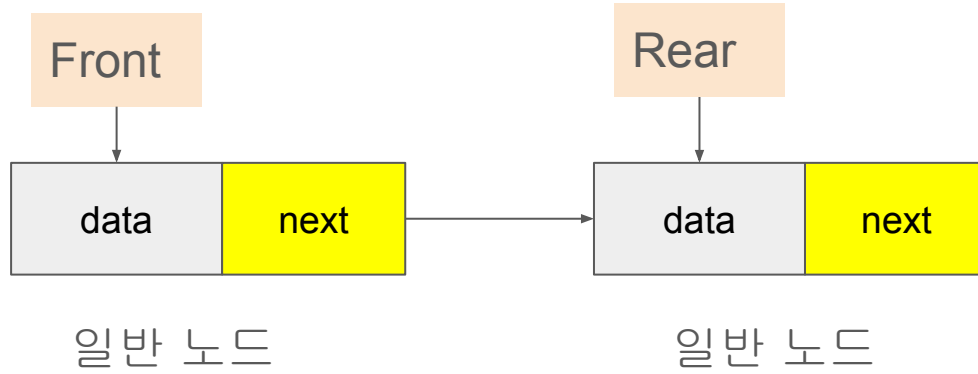


입구(뒤)

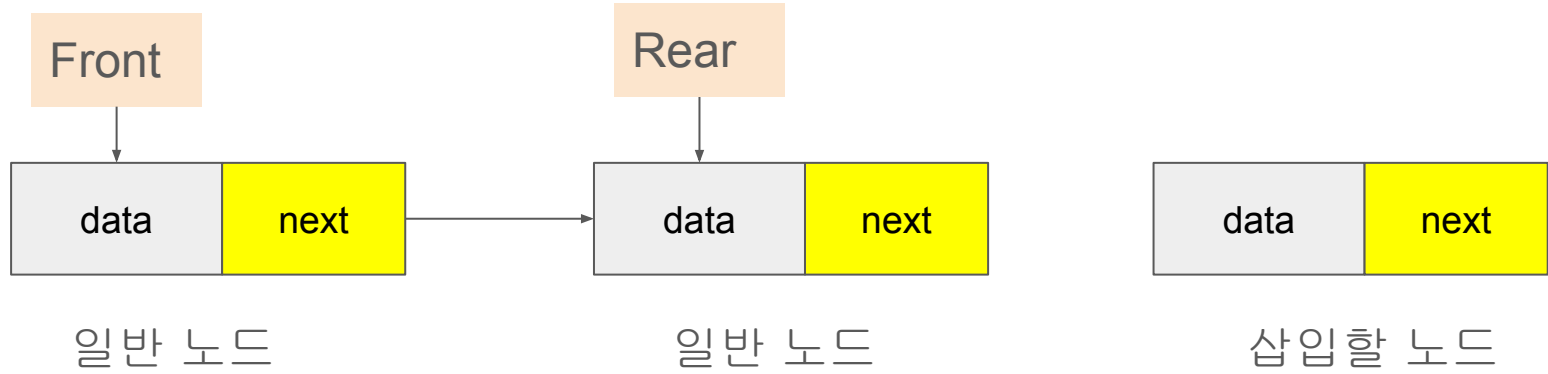
큐(Queue)

# 큐(Queue) - 링크드 리스트 구현

=> 큐 삽입 과정1



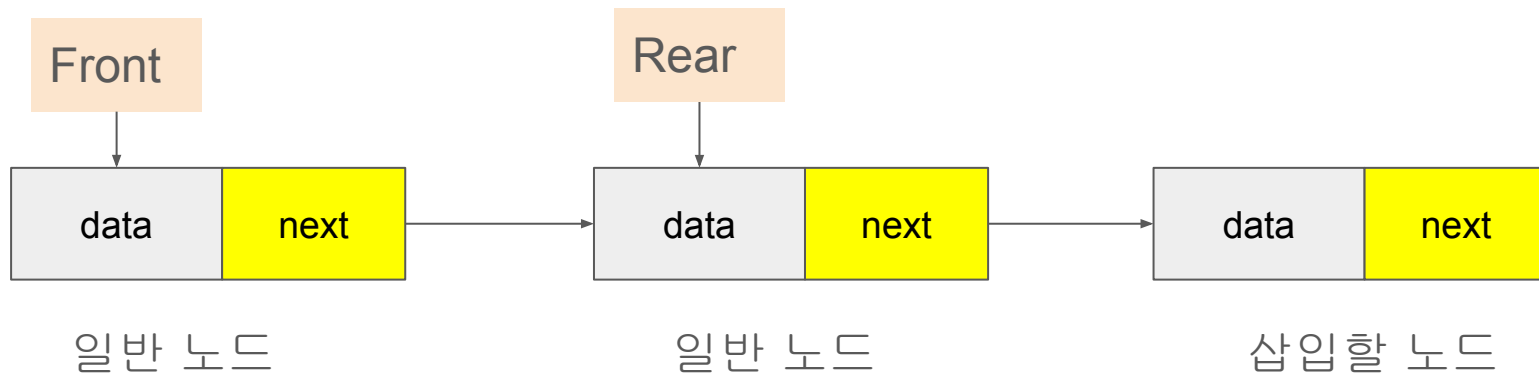
## 큐(Queue) - 링크드 리스트 구현 => 큐 삽입 과정2





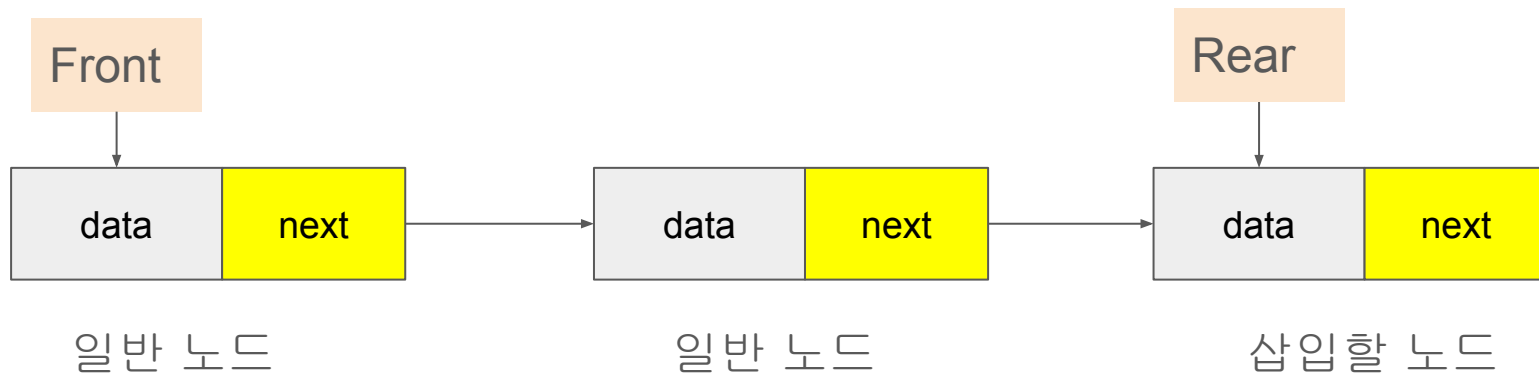
# 큐(Queue) - 링크드 리스트 구현

=> 큐 삽입 과정3



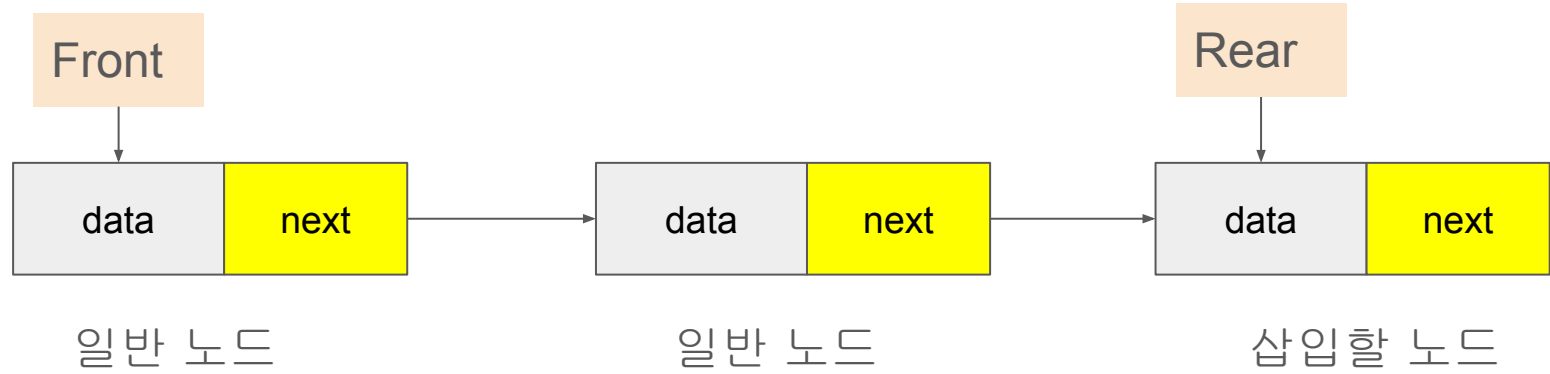
# 큐(Queue) - 링크드 리스트 구현

## => 큐 삽입 과정4



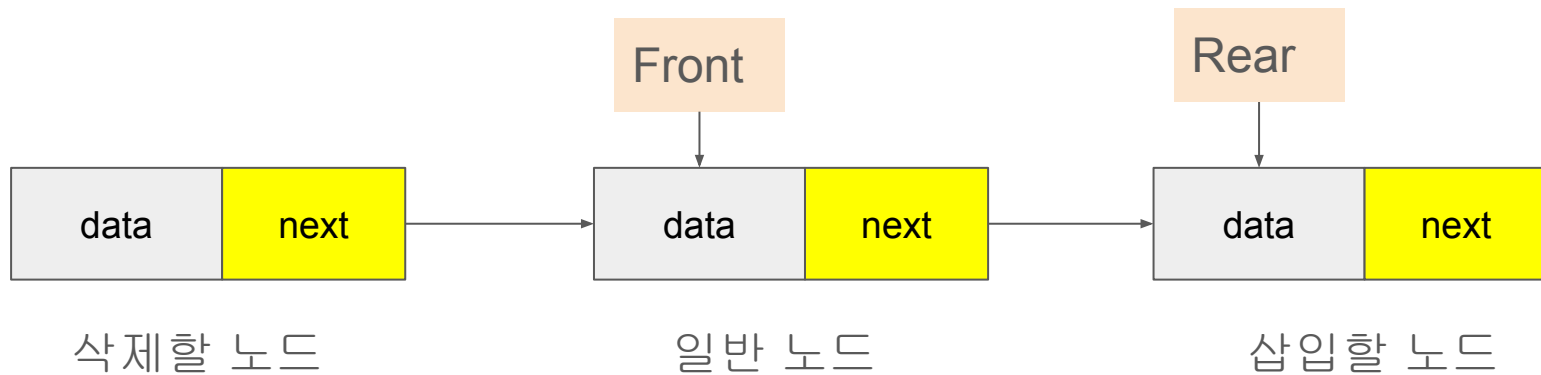
# 큐(Queue) - 링크드 리스트 구현

=> 큐 추출 과정1



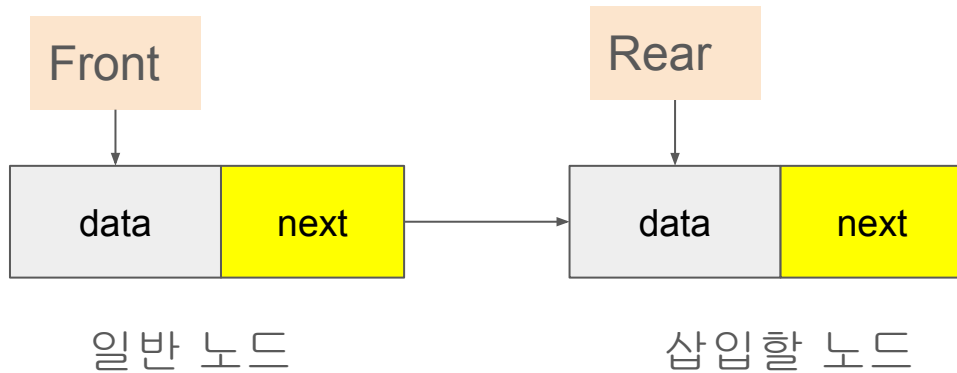
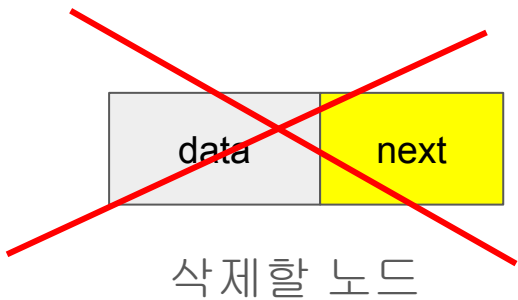
# 큐(Queue) - 링크드 리스트 구현

## => 큐 추출 과정2



# 큐(Queue) - 링크드 리스트 구현

=> 큐 추출 과정3



# 재귀 함수의 메모리 할당과 실행 recursive(3)

메모리 스택



recursive(3)  
n = 3  
call recursive(2)

recursive(2)  
n = 2  
call recursive(1)

recursive(1)  
n = 1  
call recursive(0)

recursive(0)  
n = 0  
return (호출한 데로)

반환 과정

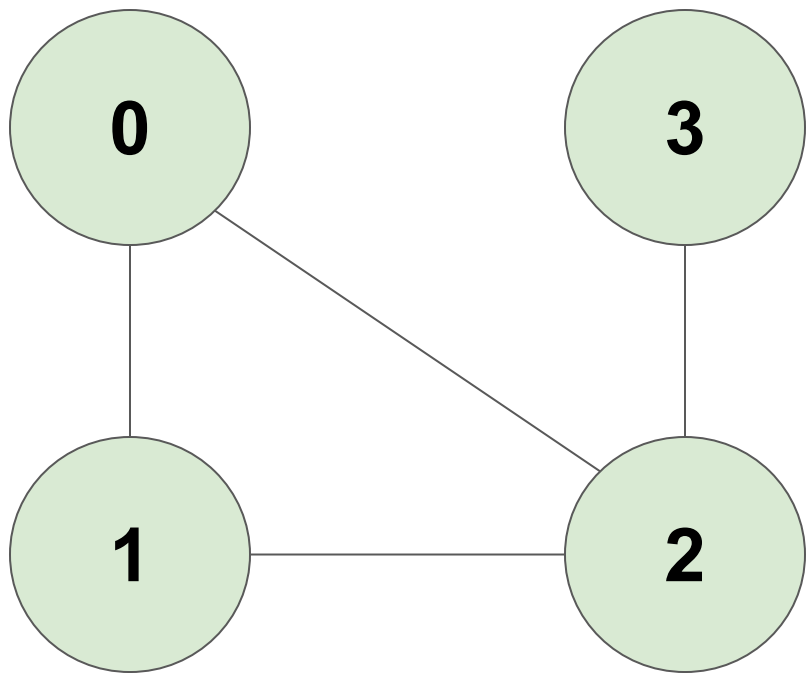


recursive(3) => main()

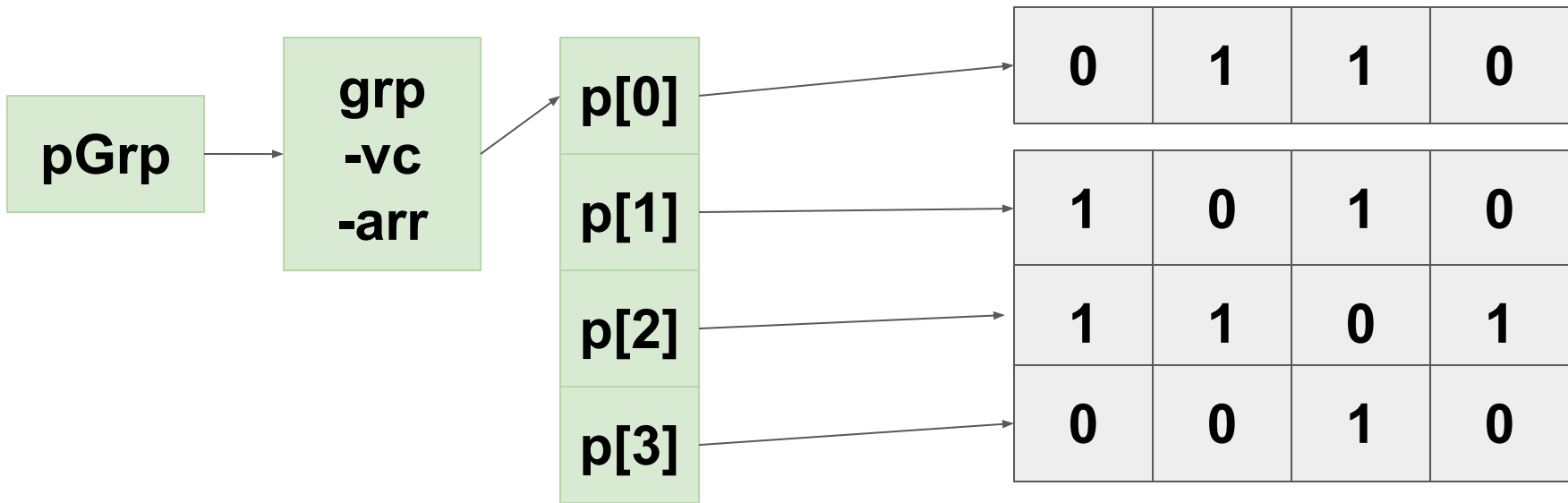
recursive(2) => recursive(3)  
출력 : 3

recursive(1) => recursive(2)  
출력 : 2

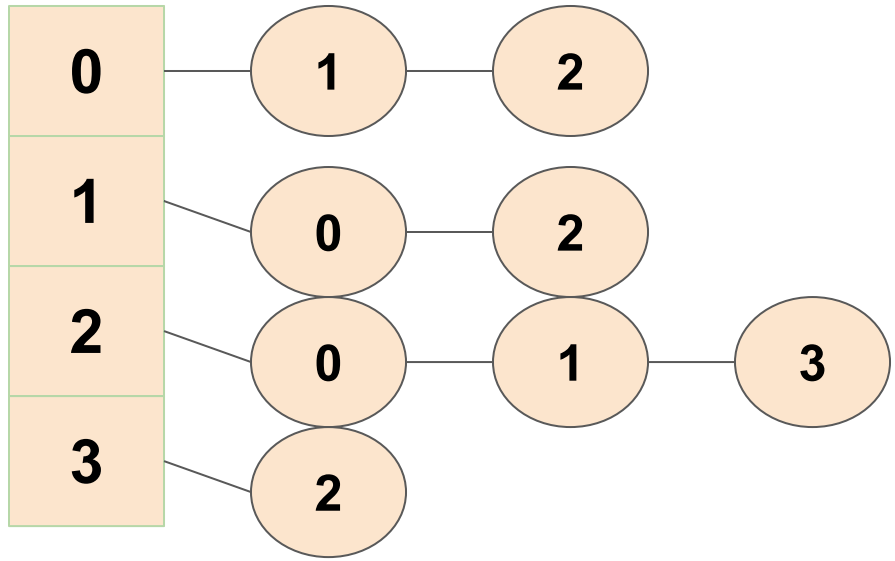
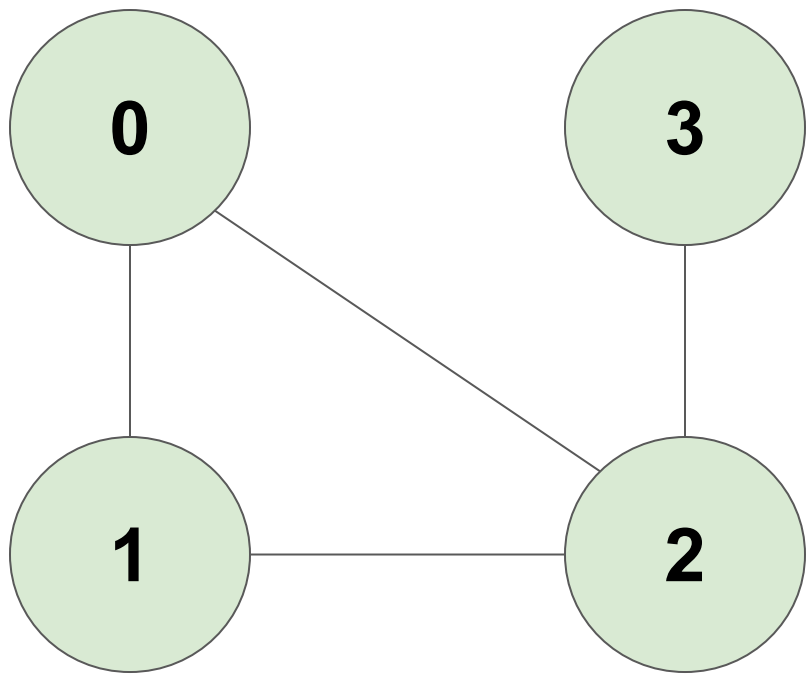
recursive(0) => recursive(1)  
출력 : 1

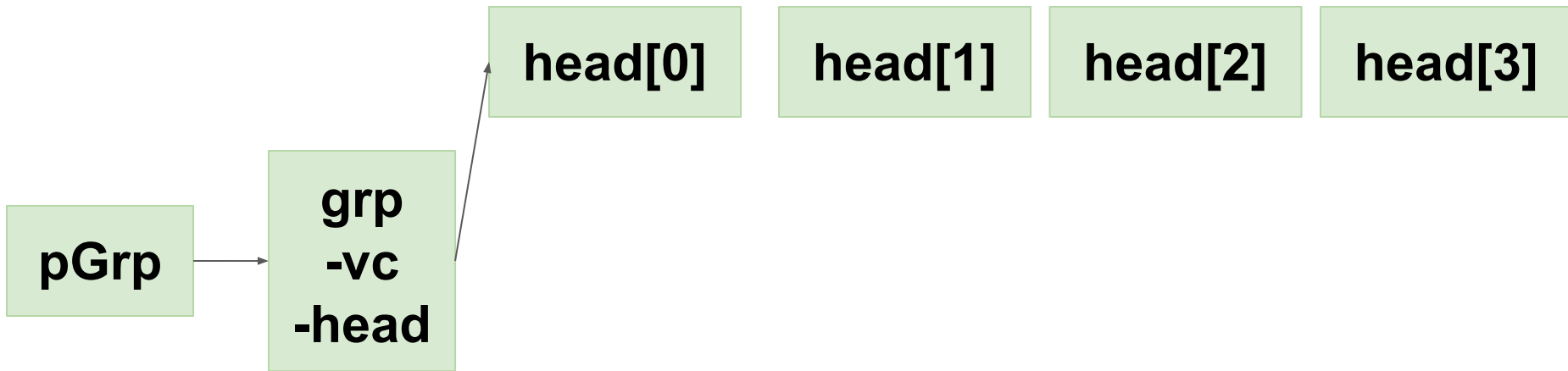


	0	1	2	3
0	0	1	1	0
1	1	0	1	0
2	1	1	0	1
3	0	0	1	0











감 사 합 니 다.