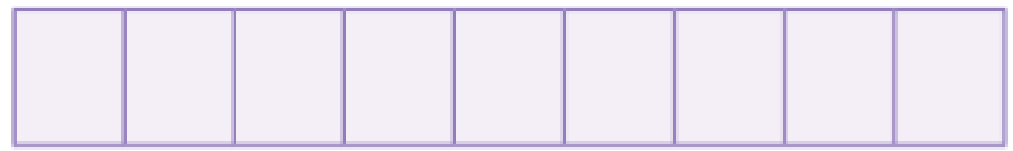




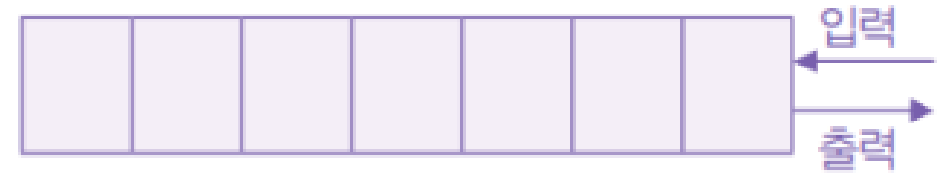
# Python Algorithm

# 선형 자료구조

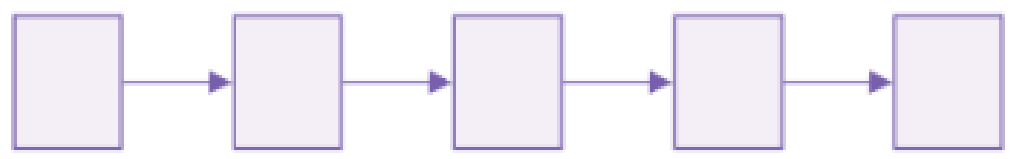
배열



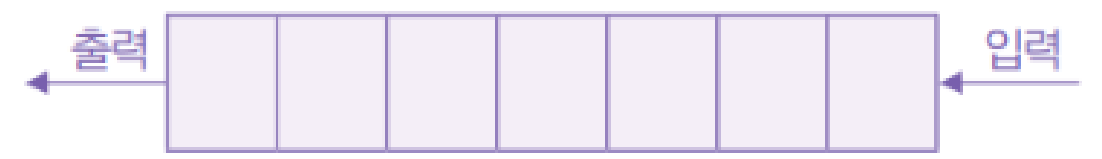
스택



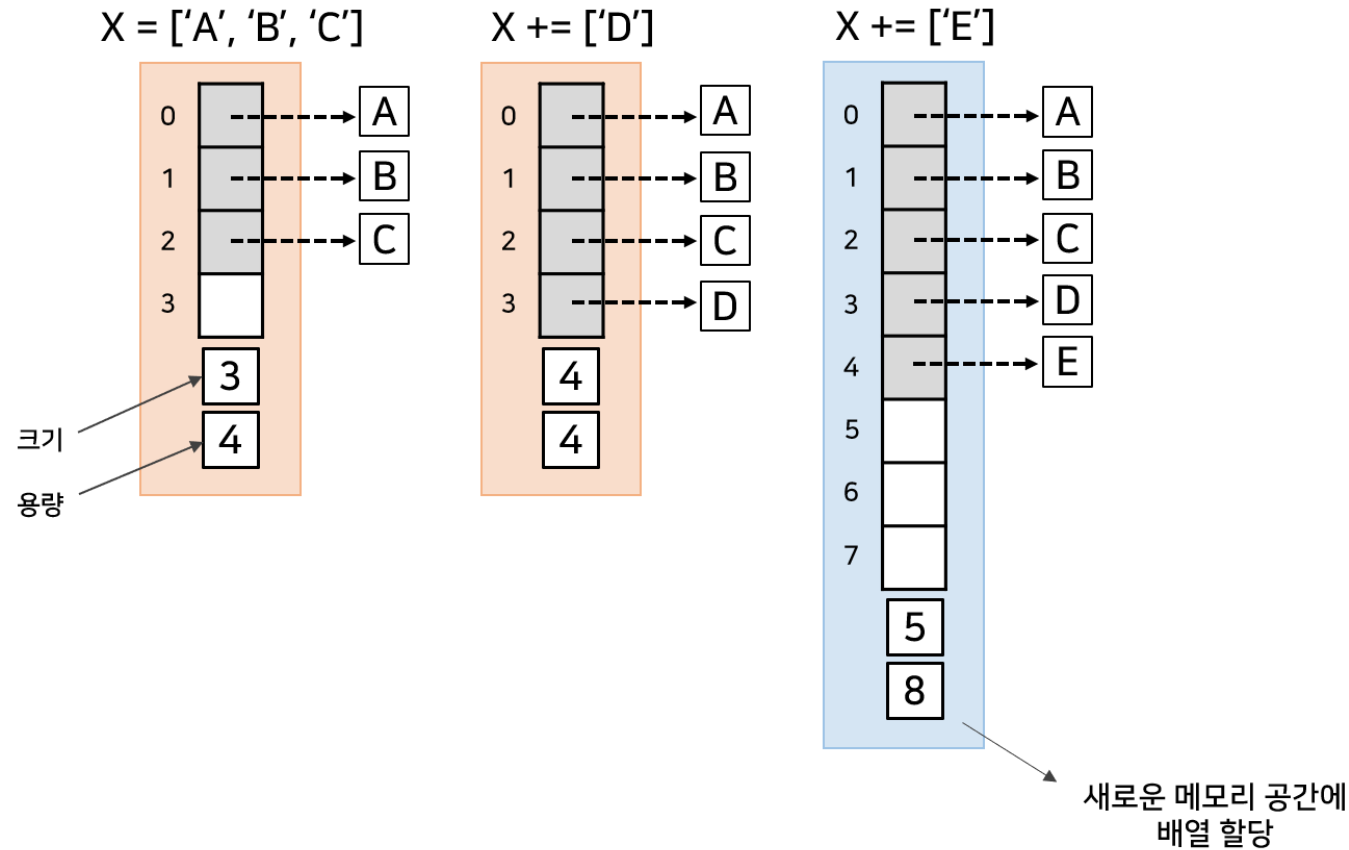
링크드 리스트



큐



# 선형 자료구조 - 배열



## 문제. 두 수의 합

<https://leetcode.com/problems/two-sum>

덧셈하여 타겟을 만들 수 있는 배열의 두 숫자 인덱스를 리턴하라.

입력	출력
<code>nums = [2, 7, 11, 15]</code> <code>target = 9</code>	<code>[0, 1]</code>

# 풀이 1. 두 수의 합

<https://leetcode.com/problems/two-sum>

## 풀이 2. 두 수의 합

<https://leetcode.com/problems/two-sum>

## 풀이 3. 두 수의 합

<https://leetcode.com/problems/two-sum>

## 풀이 4. 두 수의 합

<https://leetcode.com/problems/two-sum>



## 투 포인터 (Two Pointers)

투 포인터는 선형 자료구조에서 활용되는 문제 풀이 전략으로, 왼쪽 포인터와 오른쪽 포인터 지점을 활용합니다.

정렬 되지 않은 배열에서 특정 요소를 찾거나, Two Sum 문제와 같이 두 요소의 합이 특정 값이 되는 경우를 찾을 때, 배열 내에서 특정 조건을 만족하는 부분을 찾을 때

투 포인터 방식은 시간 복잡도를 개선하고, 보다 효율적으로 문제를 해결할 수 있도록 합니다.





# 풀이 5. 두 수의 합

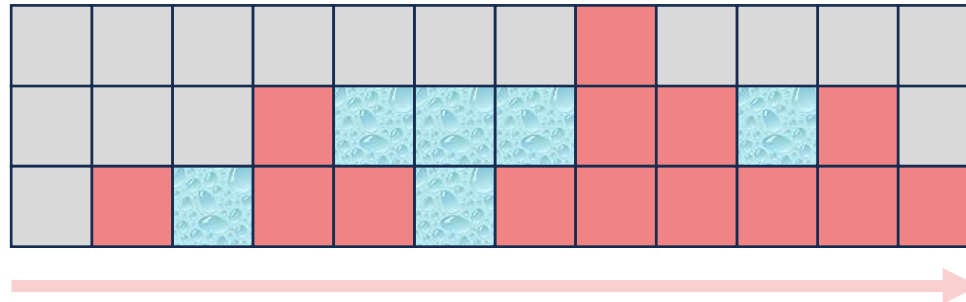
<https://leetcode.com/problems/two-sum>

높이를 입력받아 비 온 후 얼마나 많은 물이 쌓일 수 있는지 계산하라

[illegible]

# 풀이. 빗물 트래핑

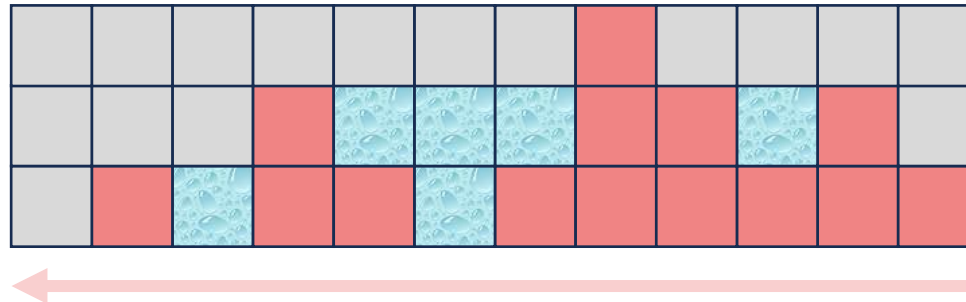
https://leetcode.com/problems/trapping-rain-water



최고 높이 0	최고 높이 1	최고 높이 1	최고 높이 2	최고 높이 2	최고 높이 2	최고 높이 2	최고 높이 3
현재 높이 0	현재 높이 1	현재 높이 0	현재 높이 2	현재 높이 1	현재 높이 0	현재 높이 1	현재 높이 3

# 풀이. 빗물 트래핑

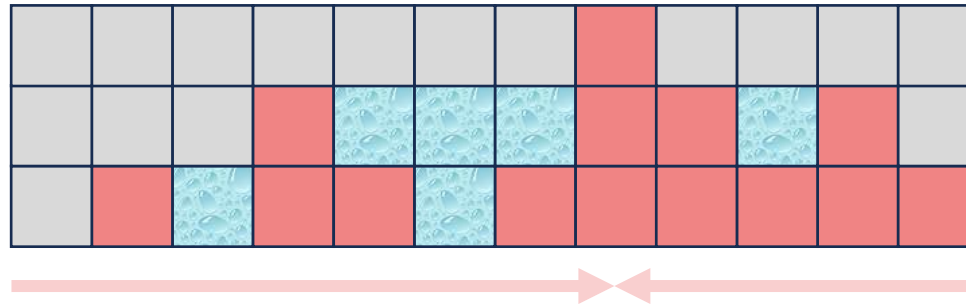
<https://leetcode.com/problems/trapping-rain-water>



최고 높이 3	최고 높이 2	최고 높이 2	최고 높이 2	최고 높이 1
현재 높이 3	현재 높이 2	현재 높이 1	현재 높이 2	현재 높이 1

# 풀이. 빗물 트래핑

<https://leetcode.com/problems/trapping-rain-water>



## 문제. 세 수의 합

<https://leetcode.com/problems/3sum>

배열을 입력받아 합으로 0을 만들 수 있는 3개의 엘리먼트를 모두 출력하라

입력	출력
<code>nums = [-1, 0, 1, 2, -1, -4]</code>	<code>[   [-1, 0, 1],   [-1, -1, 2] ]</code>

# 풀이1. 세 수의 합

<https://leetcode.com/problems/3sum>



# 풀이2. 세 수의 합

<https://leetcode.com/problems/3sum>



# 풀이3. 세 수의 합

<https://leetcode.com/problems/3sum>

## 문제. 자신을 제외한 배열의 곱

<https://leetcode.com/problems/product-of-array-except-self>

배열을 입력받아 `output[i]`가 자신을 제외한 나머지 모든 요소의 곱셈 결과가 되도록 출력하라.

단! 나눗셈을 하지 않고,  $O(n)$ 에 풀이하라.

입력	출력
<code>nums = [1, 2, 3, 4]</code>	<code>[24, 12, 8, 6]</code>

# 풀이. 자신을 제외한 배열의 곱

<https://leetcode.com/problems/product-of-array-except-self>

입력 -> [ 0번째, 1번째, 2번째, 3번째 ]

출력 -> [

1 \* 1번째 \* 2번째 \* 3번째,

1 \* 0번째 \* 2번째 \* 3번째,

1 \* 0번째 \* 1번째 \* 3번째,

1 \* 0번째 \* 1번째 \* 2번째

]

## 문제. 주식을 사고팔기 가장 좋은 시점

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock>

한 번의 거래로 낼 수 있는 최대 이익을 산출하라.

입력	출력
[7, 1, 5, 3, 6, 4]	5



# 풀이1. 주식을 사고팔기 가장 좋은 시점

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock>



## 풀이2. 주식을 사고팔기 가장 좋은 시점

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock>

# 문제. 나는 요리사다

<https://www.acmicpc.net/problem/2953>

"나는 요리사다"는 다섯 참가자들이 서로의 요리 실력을 뽐내는 티비 프로이다.  
각 참가자는 자신있는 음식을 하나씩 만들어오고, 서로 다른 사람의 음식을 점수로 평가해준다.  
점수는 1점부터 5점까지 있다.

각 참가자가 얻은 점수는 다른 사람이 평가해 준 점수의 합이다.  
이 쇼의 우승자는 가장 많은 점수를 얻은 사람이 된다.

각 참가자가 얻은 평가 점수가 주어졌을 때,  
우승자와 그의 점수를 구하는 프로그램을 작성하시오.

입력	출력
5 4 4 5	4 19
5 4 4 4	
5 5 4 4	
5 5 5 4	
4 4 4 5	

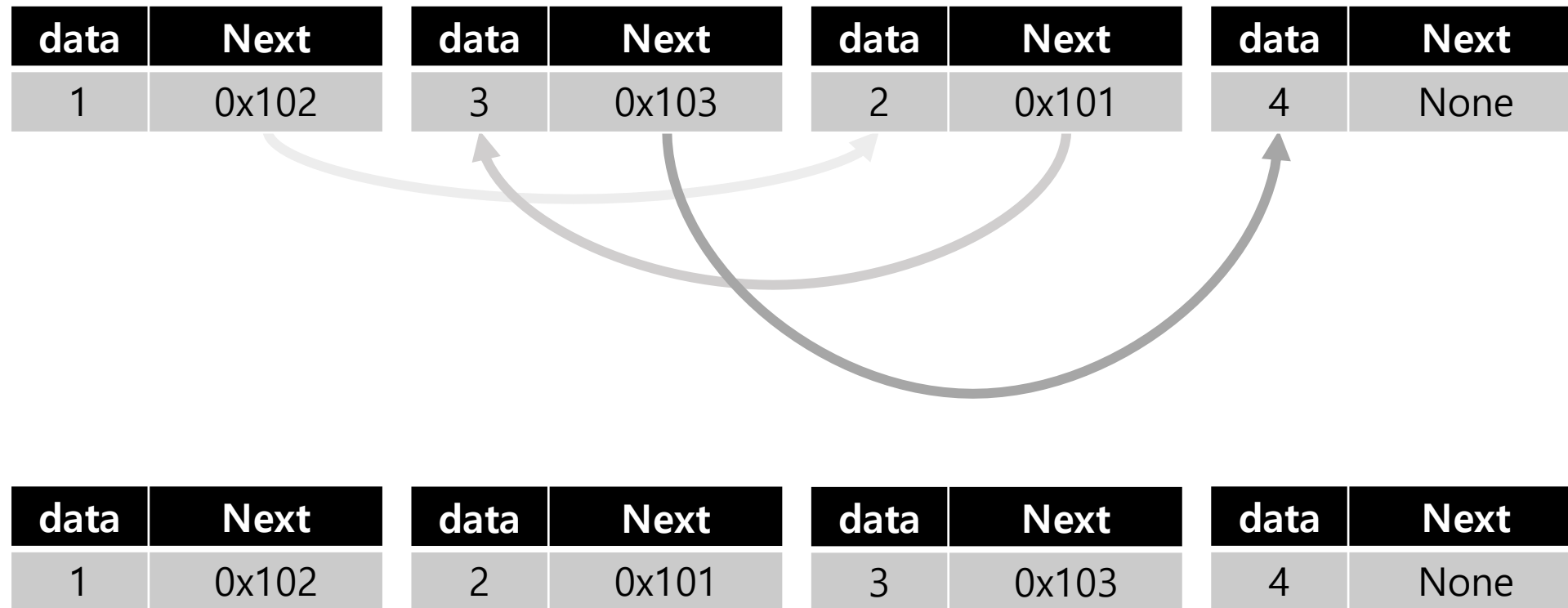




# 풀이1. 나는 요리사다

<https://www.acmicpc.net/problem/2953>

## 선형 자료구조 - 연결 리스트





# 선형 자료구조 - 연결 리스트 코드 구현하기

## 문제. 요세푸스 문제

<https://www.acmicpc.net/problem/1158>

1번부터 N번까지 N명의 사람이 원을 이루면서 앉아있고, 양의 정수  $K(\leq N)$ 가 주어진다.

이제 순서대로 K번째 사람을 제거한다.

한 사람이 제거되면 남은 사람들로 이루어진 원을 따라 이 과정을 계속해 나간다.

이 과정은 N명의 사람이 모두 제거될 때까지 계속된다.

원에서 사람들이 제거되는 순서를 (N, K)-요세푸스 순열이라고 한다.

예를 들어 (7, 3)-요세푸스 순열은  $\langle 3, 6, 2, 7, 5, 1, 4 \rangle$ 이다.

N과 K가 주어지면 (N, K)-요세푸스 순열을 구하는 프로그램을 작성하시오.

입력	출력
7 3	$\langle 3, 6, 2, 7, 5, 1, 4 \rangle$



# 풀이. 요세푸스 문제

<https://www.acmicpc.net/problem/1158>