

Python Algorithm

알고리즘이란..?

문제를 풀기 위해 특정한 연산을 수행하는 규칙의 묶음

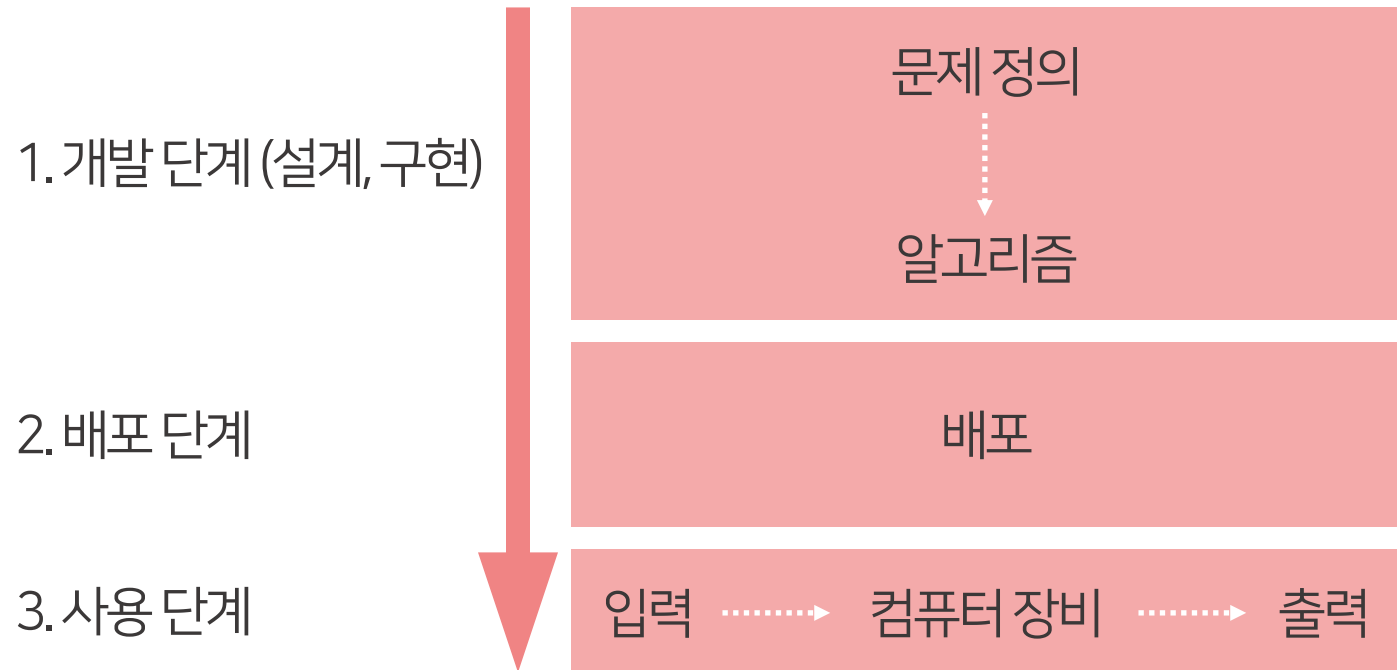
- *“알고리즘은 어떤 초기 조건의 집합이 주어졌을 때, 인식 가능한 일련의 종료 조건을 가진 특정 목표를 달성하기 위해서 정해진 순서에 따라 수행될 수 있는 명료한 요구사항으로 구성된 유한한 집합이다.”*
- 알고리즘을 설계하는 것은 현실 세계의 문제를 해결하기 위해, 사용될 수 있는 가장 효율적인 방법으로 수학적 레시피를 만드는 것!

온라인 테스트 플랫폼

- 해커랭크 (<https://www.hackerrank.com>)
- 코딜리티 (<https://www.codility.com>)
- 리모트인터뷰 (<https://www.remoteinterview.io>)
- 프로그래머스 (<https://programmers.co.kr>)

- 리트코드(<https://leetcode.com>)
- 백준 (<https://www.acmicpc.net>)

알고리즘의 단계



파이썬 문법 - 들여쓰기

```
def function_name (var_one, var_two,  
                    var_three, var_four):  
    print(var_one)
```

```
def function_name (  
    var_one, var_two, var_three,  
    var_four):  
    print(var_one)
```

파이썬 문법 - 네이밍 컨벤션

Camel case

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "email": "john.smith@example.com",  
  "createdAt": "2021-02-20T07:20:01",  
  "updatedAt": "2021-02-20T07:20:01",  
  "deletedAt": null  
}
```

Snake case

```
{  
  "first_name": "John",  
  "last_name": "Smith",  
  "email": "john.smith@example.com",  
  "created_at": "2021-02-20T07:20:01",  
  "updated_at": "2021-02-20T07:20:01",  
  "deleted_at": null  
}
```

파이썬 문법 - 타입 힌트

```
a: str = "1"  
b: int = 1  
c: float = 1.0
```

```
def fn(a:int) -> bool:  
    ...
```

파이썬 문법 - 타입 힌트

```
pip install mypy
```

```
mypy 파일명.py
```

```
$ mypy test.py
```

```
test.py:7: error: Incompatible return value type (got "int", expected "bool") [return-value]  
Found 1 error in 1 file (checked 1 source file)
```

```
$ mypy test.py
```

```
Success: no issues found in 1 source file
```


파이썬 문법 - 리스트 컴프리헨션

기존 리스트를 기반으로 새로운 리스트를 만들어내는 구문

```
# 홀수인 경우, 2를 곱해 리스트를 만들기  
list = [n * 2 for n in range(1, 10 + 1) if n % 2 == 1]
```

```
# {키: 키의 제곱} 딕셔너리 만들기  
double_dict = {x: x**2 for x in range(1, 5 + 1)}
```

파이썬 문법 - 제너레이터

루프의 반복 동작을 제어할 수 있는 루틴 형태

```
def get_natural_number():  
    n = 0  
    while True:  
        n += 1  
        yield n  
  
n_num = get_natural_number()  
  
for _ in range(0, 100):  
    print(next(n_num))
```

파이썬 문법 - 제너레이터

여러 타입의 데이터를 하나의 함수에서 생성하는 것도 가능

```
def get_many_type_data():  
    while True:  
        yield 1  
        yield '문자열'  
        yield True  
  
data = get_many_type_data()  
  
for _ in range(0, 10):  
    print(next(data))
```

파이썬 문법 - range()

제너레이터 방식을 활용하는 대표적 함수로는 range()가 있습니다.

```
a = [n for n in range(1000000)]  
b = range(1000000)  
  
print(len(a) == len(b))
```

```
import sys  
  
print(sys.getsizeof(a))  
print(sys.getsizeof(b))
```

파이썬 문법 - enumerate()

enumerate 함수는 list, set, tuple 등을 인덱스를 포함한 enumerate 객체로 리턴합니다.

```
a = ["red", "orange", "yellow",  
     "green", "blue", "navy", "purple"]  
  
enum_a = enumerate(a)  
  
print(enum_a)  
print(list(enum_a))
```

파이썬 문법 - enumerate()

enumerate 함수는 list, set, tuple 등을 인덱스를 포함한 enumerate 객체로 리턴합니다.

```
a = ["red", "orange", "yellow",  
     "green", "blue", "navy", "purple"]
```

리스트 a에서 인덱스와 값을 같이 함께 출력하려면?

```
for i in range(len(a)):  
    print(i, a[i])
```

파이썬 문법 - enumerate()

enumerate 함수는 list, set, tuple 등을 인덱스를 포함한 enumerate 객체로 리턴합니다.

```
a = ["red", "orange", "yellow",  
     "green", "blue", "navy", "purple"]  
  
# 리스트 a에서 인덱스와 값을 같이 함께 출력하려면?  
  
i = 0  
for v in a:  
    print(i, v)  
    i += 1
```

파이썬 문법 - enumerate()

enumerate 함수는 list, set, tuple 등을 인덱스를 포함한 enumerate 객체로 리턴합니다.

```
a = ["red", "orange", "yellow",  
     "green", "blue", "navy", "purple"]
```

리스트 a에서 인덱스와 값을 같이 함께 출력하려면?

```
for i, v in enumerate(a):  
    print(i, v)
```


파이썬 문법 - locals()

locals 함수는 내부의 모든 로컬 변수를 출력해주기 때문에 디버깅에 많은 도움을 줍니다.

```
...  
print(locals())  
  
import pprint  
pprint.pprint(locals())
```

잘 짤 코드란?

- 시간복잡도 : 작업을 얼마나 더 빠르게 처리하는지
- 공간복잡도 : 실행하면서 메모리를 얼마나 사용하는지
- 가독성 : 알아보기 쉽고, 한 번에 이해할 수 있도록 보기 좋게 작성되었는지



Big-O 알고리즘

Big-O 알고리즘은 점근적 실행 시간을 표기할 때, 가장 널리 쓰이는 수학적 표기법 중 하나로 알고리즘의 효율성을 분석하는데 매우 유용하게 사용됩니다.



1MB



1,024TB

Big-O 알고리즘

```
def sum(n):  
    total = 0  
    for i in range(1, n+1):  
        total += i  
    return total
```

...(ㄱ) 1회 실행

...(ㄴ) 리스트의 크기 n

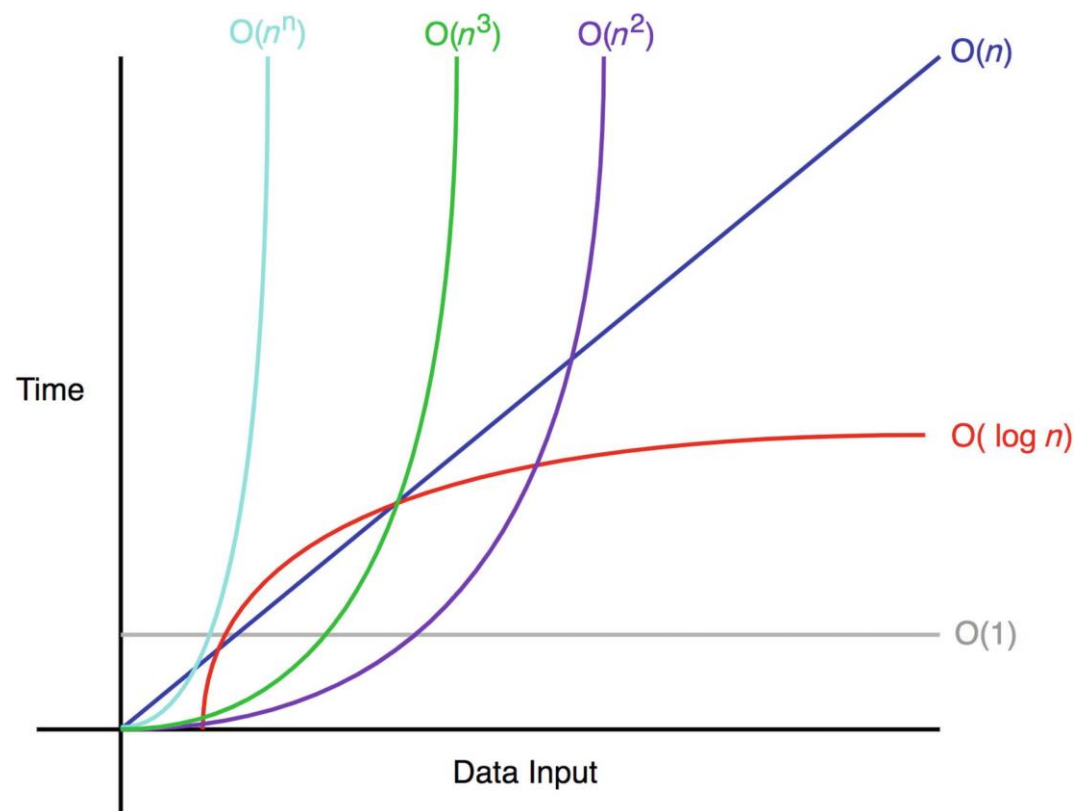
...(ㄷ) 산술연산 1회, 대입연산 1회 (2회)

시간 복잡도는 $2n+1$ 이 되고, Big-O 표기법에 따르면 $O(n)$ 이 됩니다.

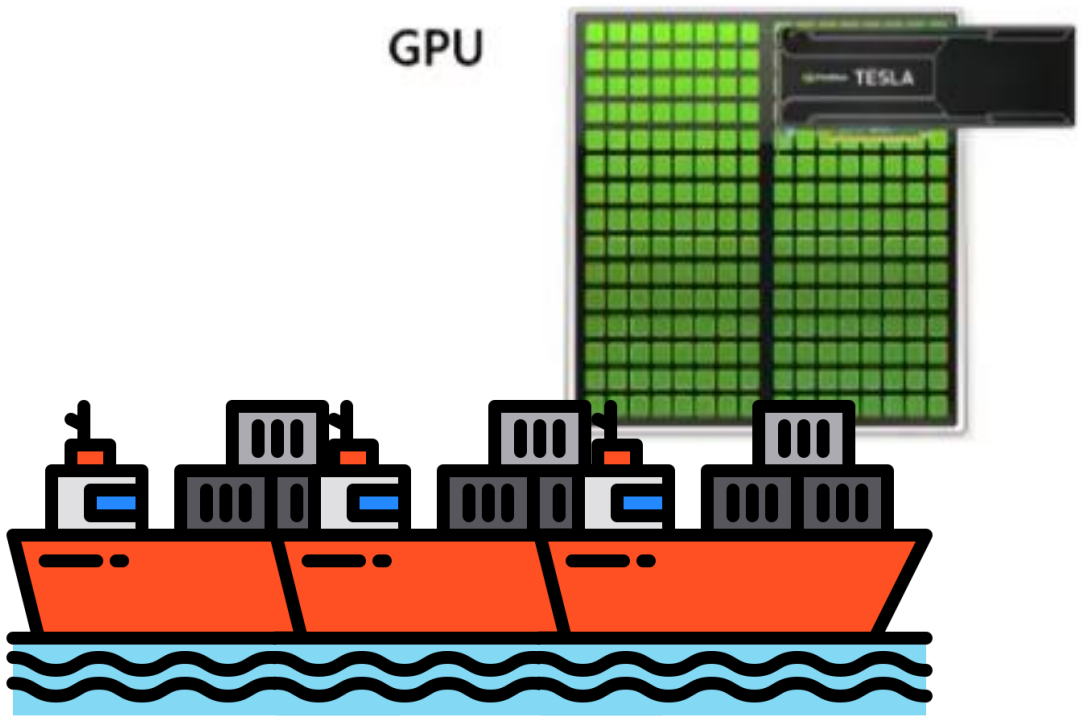
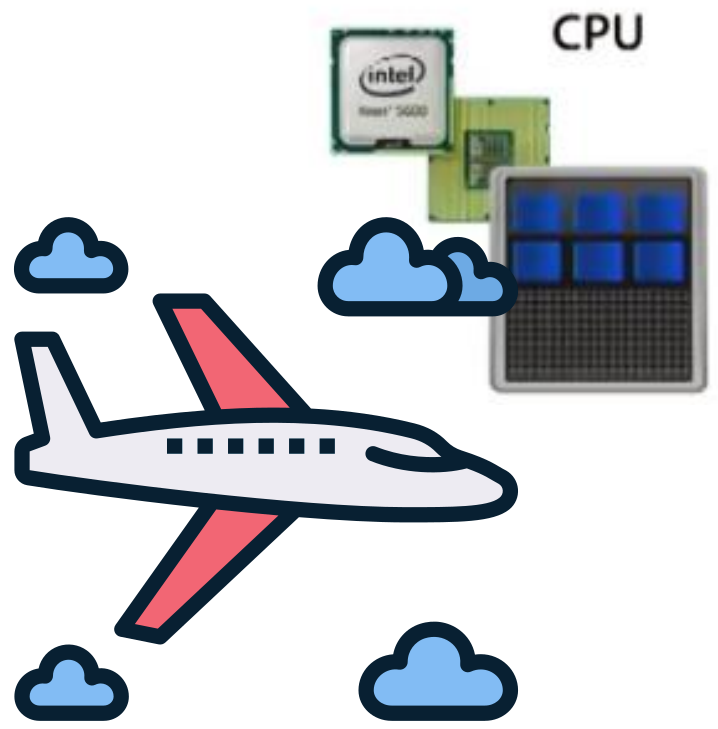
Big-O 알고리즘

```
def sum(n):  
    return n * (n+1) // 2 ... (ㄱ) 산술연산 3회
```

시간 복잡도는 3이 되고, Big-O 표기법에 따르면 $O(1)$ 이 됩니다.



병렬화



Big-O 알고리즘

시간 복잡도	설명
$O(1)$	입력값이 아무리 커도 실행 시간이 일정한 최고의 알고리즘 대표적인 알고리즘으로 해시 테이블의 조회 및 삽입이 해당됩니다.
$O(\log n)$	매우 큰 입력 값에도 영향을 받는 정도가 미미하므로, 비교적 빠르게 실행되는 알고리즘 대표적으로는 이진 검색이 이에 해당합니다.
$O(n)$	입력값만큼 실행 시간에 영향을 받는 선형 알고리즘 정렬되지 않은 리스트에서 최대값 또는 최소값을 찾는 경우가 이에 해당합니다.
$O(n \log n)$	대부분의 효율 좋은 정렬 알고리즘이 여기에 해당됩니다.
$O(n^2)$	입력 크기의 제곱에 비례하여 실행 시간이 증가하는 알고리즘 버블 정렬과 같은 비효율적인 정렬 알고리즘이 여기에 해당합니다.
$O(2^n)$	입력 크기가 조금만 증가해도 알고리즘의 실행 시간이 급격하게 늘어나며, 하노이 탑과 같은 재귀 알고리즘이 여기에 해당합니다.
$O(n!)$	매우 비효율적인 알고리즘으로 작은 입력에도 실행 시간이 매우 길기 때문에, 실제 상황에서는 사용되지 않습니다.

자주 사용하는 자료 구조의 시간 복잡도

자료구조	탐색	삽입	삭제
배열	$O(n)$	$O(n)$	$O(n)$
정렬된 배열	$O(\log n)$	$O(n)$	$O(n)$
연결 리스트	$O(n)$	$O(1)$	$O(1)$
스택/큐	$O(n)$	$O(1)$	$O(1)$
해시	$O(1)$	$O(1)$	$O(1)$
이진트리	$O(\log n)$	$O(\log n)$	$O(\log n)$

시간 복잡도 줄이기

- 문자열 합치기: + 연산자를 사용하지 않고, "".join()을 사용
- 조건문 연산 줄이기:
다중조건문의 경우 선후 관계를 고려
조건문이 중첩되는 경우, 효과적인 경우의 수를 계산
- 표준 라이브러리 활용하기

문제. 달팽이는 올라가고 싶다

<https://www.acmicpc.net/problem/2869>

땅 위에 달팽이가 있다. 이 달팽이는 높이가 V미터인 나무 막대를 올라갈 것이다.
달팽이는 낮에 A미터 올라갈 수 있다. 하지만 밤에 잠을 자는 동안 B미터 미끄러진다.
또, 정상에 올라간 후에는 미끄러지지 않는다.

달팽이가 나무 막대를 모두 올라가려면, 며칠이 걸리는지 구하는 프로그램을 작성하시오.

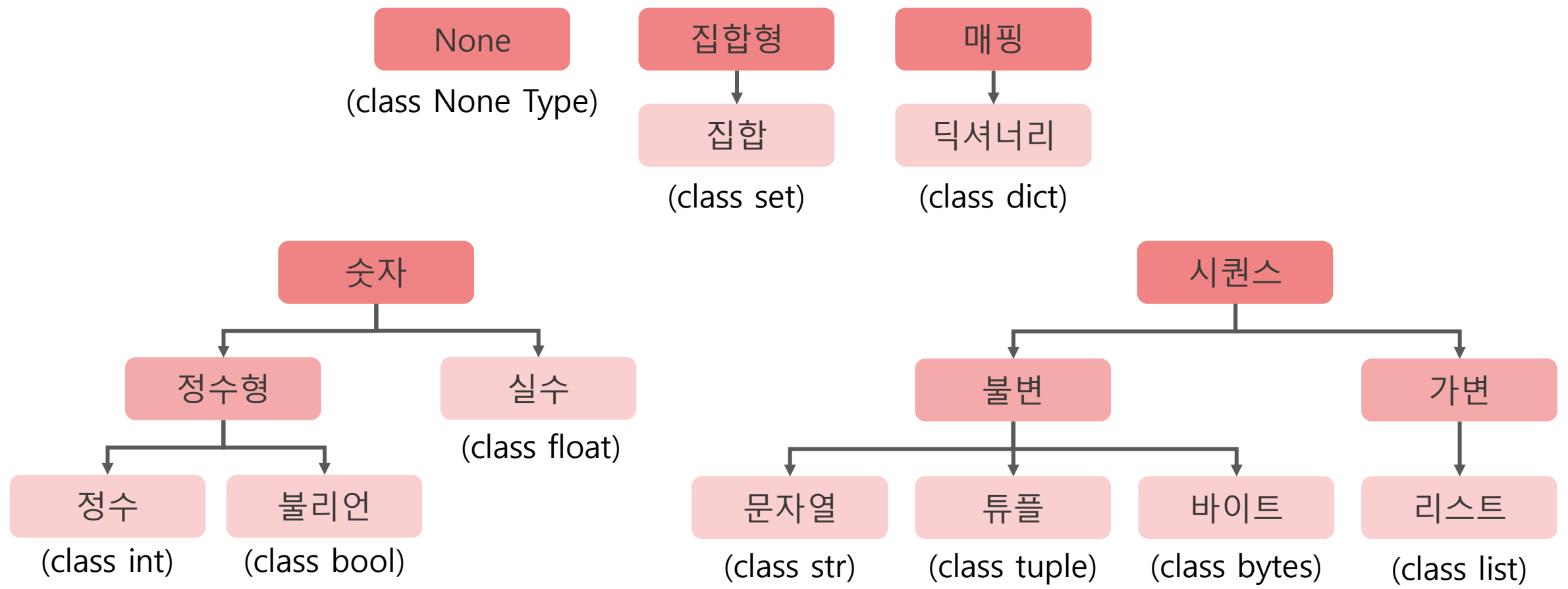
입력(ABV)	출력
215	4
입력	출력
516	2
입력	출력
100 99 10000000000	9999999901



풀이. 달팽이는 올라가고 싶다

<https://www.acmicpc.net/problem/2869>

파이썬의 자료형



is, ==

==는 값을 비교하는 반면에, is는 id()의 값, 다시 말해 메모리의 주소를 비교하게 됩니다.

```
import copy

a = [1, 2, 3]
b = copy.deepcopy(a)

# a와 b는 같은 값을 가지지만, 완전히 다른 객체로 복사됨
print(a == b)
print(a is b)
```

리스트 연산의 시간 복잡도

연산	시간 복잡도
len(a)	$O(1)$
a[i]	$O(1)$
a[i:j]	$O(k)$
element in a	$O(n)$
a.count(element)	$O(n)$
a.index(element)	$O(n)$
a.append(element)	$O(1)$

연산	시간 복잡도
a.pop()	$O(1)$
a.pop(0)	$O(n)$
del a[i]	$O(n)$
a.sort()	$O(n \log n)$
min(a)	$O(n)$
max(a)	$O(n)$
a.reverse()	$O(n)$

딕셔너리 연산의 시간 복잡도

연산	시간 복잡도
len(a)	$O(1)$
a[key]	$O(1)$
a[key] = value	$O(1)$
key in a	$O(1)$

문제. 오픈채팅방 - 2020 카카오 개발자 신입 공개채용 1차

<https://programmers.co.kr/learn/courses/30/lessons/42888>

카카오톡 오픈채팅방에서는 친구가 아닌 사람들과 대화를 할 수 있는데, 본래 닉네임이 아닌 가상의 닉네임을 사용하여 채팅방에 들어갈 수 있다.

채팅방에 들어오고 나가거나, 닉네임을 변경한 기록이 담긴 문자열 배열 record가 매개변수로 주어질 때, 모든 기록이 처리된 후, 최종적으로 방을 개설한 사람이 보게 되는 메시지를 문자열 배열 형태로 return 하도록 solution 함수를 완성하라.

입력	출력
<pre>["Enter uid1234 Muzi", "Enter uid4567 Prodo", "Leave uid1234", "Enter uid1234 Prodo", "Change uid4567 Ryan"]</pre>	<pre>["Prodo님이 들어왔습니다.", "Ryan님이 들어왔습니다.", "Prodo님이 나갔습니다.", "Prodo님이 들어왔습니다."]</pre>



풀이. 오픈채팅방 - 2020 카카오 개발자 신입 공개채용 1차

<https://programmers.co.kr/learn/courses/30/lessons/42888>

딕셔너리 - defaultdict

```
a = collections.defaultdict(int)
a['A'] = 5
a['B'] = 4
print(a)
a['C'] += 1
```

딕셔너리 - Counter

```
a = [1, 2, 3, 4, 5, 5, 5, 6, 6]
b = collections.Counter(a)

print(b)

print(b.most_common(2))
```



문제. 숫자의 합

<https://www.acmicpc.net/problem/11720>

N개의 숫자가 공백없이 쓰여있다.

이 숫자를 모두 합해서 출력하는 프로그램을 작성하시오.

입력	출력
5 12345	15

풀이. 숫자의 합

<https://www.acmicpc.net/problem/11720>

문제. 팰린드롬 (회문)

<https://leetcode.com/problems/valid-palindrome>

주어진 문자열이 팰린드롬인지 확인하라. 대소문자를 구분하지 않으며, 영문자와 숫자만을 대상으로 한다.

입력	출력
"A man, a plan, a canal : Panama"	true
입력	출력
"race a car"	false

풀이 1. 팰린드롬 (회문) - 리스트로 변환

<https://leetcode.com/problems/valid-palindrome>

풀이 2. 팰린드롬 (회문) - 데크 자료형을 이용

<https://leetcode.com/problems/valid-palindrome>

풀이 3. 팰린드롬 (회문) - 슬라이싱 사용

<https://leetcode.com/problems/valid-palindrome>

문제. 문자열 뒤집기

<https://leetcode.com/problems/reverse-string>

문자열을 뒤집는 함수를 작성하라.

입력값은 문자 배열이며, 리턴 없이 리스트 내부를 직접 조작하라

입력 ["h", "e", "l", "l", "o"]	출력 ['o', 'l', 'l', 'e', 'h']
입력 ["H", "a", "n", "n", "a", "h"]	출력 ['h', 'a', 'n', 'n', 'a', 'H']

풀이 1. 문자열 뒤집기 - 투 포인터를 이용한 스왑

<https://leetcode.com/problems/reverse-string>



풀이 2. 문자열 뒤집기 - 파이썬 기능 활용

<https://leetcode.com/problems/reverse-string>

문제. 로그 파일 재정렬

<https://leetcode.com/problems/reorder-data-in-log-files>

로그를 재정렬하라. 기준은 다음과 같다.

1. 로그의 가장 앞 부분은 식별자다.
2. 문자로 구성된 로그가 숫자 로그보다 앞에 온다.
3. 식별자는 순서에 영향을 끼치지 않지만, 문자가 동일할 경우 식별자 순으로 한다.
4. 숫자 로그는 입력 순서대로 한다.

입력

```
["dig1 8 1 5 1","let1 art can","dig2 3 6","let2 own kit dig","let3 art zero"]
```

출력

```
["let1 art can","let3 art zero","let2 own kit dig","dig1 8 1 5 1","dig2 3 6"]
```



풀이. 로그 파일 재정렬

<https://leetcode.com/problems/reorder-data-in-log-files>

람다 표현식

```
s = ['2 C', '1 A', '4 D', '3 B']  
s.sort()  
print(s)
```

```
def func(x):  
    return x.split()[1], x.split()[0]  
s.sort(key=func)  
print(s)
```

```
s.sort(key=lambda x: (x.split()[1], x.split()[0]))  
print(s)
```


람다 표현식

```
s = ['2 C', '1 A', '4 D', '3 B']  
print(sorted(s))
```

```
def func(x):  
    return x.split()[1], x.split()[0]  
print(sorted(s, key=func))
```

```
print(sorted(s, key=lambda x: (x.split()[1], x.split()[0])))
```

문제. 가장 흔한 단어

<https://leetcode.com/problems/most-common-word>

금지된 단어를 제외한 가장 흔하게 등장하는 단어를 출력하라.
대소문자 구분을 하지 않으며, 구두점(마침표, 쉼표) 또한 무시한다.

입력

```
paragraph = "Bob hit a ball, the hit BALL flew far after it was hit."  
banned = ["hit"]
```

출력

```
"ball"
```

풀이. 가장 흔한 단어

<https://leetcode.com/problems/most-common-word>

문제. 그룹 애너그램

<https://leetcode.com/problems/group-anagrams>

문자열 배열을 받아 애너그램 단위로 그룹핑하라.

입력

```
["eat", "tea", "tan", "ate", "nat", "bat"]
```

출력

```
[  
  ["ate", "eat", "tea"],  
  ["nat", "tan"],  
  ["bat"]  
]
```

풀이. 그룹 애너그램

<https://leetcode.com/problems/group-anagrams>

문제. 가장 긴 팰린드롬 부분 문자열

<https://leetcode.com/problems/longest-palindromic-substring>

가장 긴 팰린드롬 부분 문자열을 출력하라

입력 "babad"	출력 "bab"
입력 "cbbd"	출력 "bb"



풀이. 가장 긴 팰린드롬 부분 문자열

<https://leetcode.com/problems/longest-palindromic-substring>

문제. 백대열

<https://www.acmicpc.net/problem/14490>

대열이는 육제의 친구이다.

- "야 백대열을 약분하면 뭔지 알아?"
- "??"
- "십대일이야~ 하하!"

n 과 m 이 :을 사이에 두고 주어진다. 두 수를 최대한으로 약분하여 출력한다.

입력	출력
100:10	10:1



문제. 백대열

<https://www.acmicpc.net/problem/14490>