

# Computing Foundations for Data Science

## HW 12

**제출기한: 2022/04/28 02:00PM**

### 주의사항

- 코드를 Jupyter Notebook 에서 작성하였더라도 python 파일(.py)로 제출할 것.
- 함수가 의도한 값을 return 하는지를 확인할 것. (print 와 혼동하지 말 것)
- 파일명은 P1.py ~ P4.py 를 유지하고, 해당 파일들을 HW#\_학번\_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 Keondo Park 이라면 **HW12\_2020\_12345\_KeondoPark.zip** 으로 압축하여 제출.
- 예시로 제시한 입력 값 외에도 조교가 임의로 생성한 입력 값으로도 코드가 잘 실행되는지 테스트할 예정.
- 뼈대 코드의 함수 이름 및 매개변수(parameter)는 변경하지 말 것.
- **기본 모듈만** 이용
- 이번 과제는 hashing 에 대한 과제이다. 다음 모든 문제들은 **time complexity  $O(n)$** 으로 해결할 수 있다. 따라서 다음과 같이 두 부류로 test case 를 구성할 것이다.
  - 1) 입력 리스트의 원소의 개수가 적고, return 값만 맞으면 정답
  - 2) 입력 리스트의 원소의 개수가 충분히 많은 ( $10^7$  이상) 경우. **실행 시간이 특정 시간을 넘어가면 return 값에 상관없이 틀림**여기서 **특정 시간**은  $O(n)$ 으로 실행하는 시간보다는 충분히 크고,  $O(n^2)$ 으로 실행하는 시간 보다는 충분히 작은 시간으로 설정할 것이다.  $O(n \log n)$ 으로 해결했을 경우, 정답을 보장할 수 없다. **set 또는 dictionary** 를 이용해서  $O(n)$ 으로 해결해보자.

## 문제 1.

0 이상 정수로 이루어진 리스트 `nums`와 0 이상 정수 `k` 를 입력으로 받는다. 리스트의 원소를 둘씩 짝지으려고 하는데, 각각의 쌍의 합이 모두 `k`로 나누어지도록 짝을 지으려 한다. 가능하면 `True`, 불가능하면 `False` 를 `return` 하는 함수를 구현하시오.

(리스트의 길이는 2 이상 짝수이다)

## 예시

```
>>>P1([123, 36, 54, 28, 39, 28], 2)
True
```

설명: (123, 39), (36, 54), (28, 28) 이렇게 짝을 지으면 각각의 쌍의 합이 모두 2로 나눈다. 짝을 짓는 방법은 여러 가지가 있을 수 있다.

```
>>>P1([123, 36, 54, 28, 39, 28], 3)
False
```

설명: 각각의 쌍의 합이 3으로 나누어지도록 나눌 수 없다.

```
>>>P1([3,7,6,5,4,5], 5)
True
```

설명: (3, 7), (6, 4), (5, 5) 이렇게 짝을 지으면 가능하다.

## 문제 2.

0 과 1 로만 이루어진 리스트 `nums` 를 입력으로 받는다. 0 과 1 의 개수가 같은 부분 리스트의 길이의 최대값을 `return` 하는 함수를 구현하시오. 부분 리스트란,  $nums[i:j]$ ,  $0 \leq i \leq j \leq len(nums)$  와 같이 리스트의 연속된 일부분을 뜻한다. 그러한 부분 리스트가 없으면 0 을 `return` 하면 된다. (리스트의 길이는 1 이상이다.)

### 예시

$$\text{sum} = \begin{cases} 1 & : 0 \\ 2 & : 1 \end{cases}$$

이 index

```
>>> P2([1,1,0,1,0,1])
4
```

now = 1 → 2 → 1 → 2 → 1 → 2  
i = 0 1 2 3 4 5

설명: `[1,1,0,1,0,1]` 0 과 1 의 개수가 같은 부분 리스트 중에서 길이의 최대값은 4 이다. 길이가 최대인 부분 리스트는 여러 개가 있을 수 있고, 길이의 최대값만 `return` 하면 된다.

$$\text{ans} = 2 \quad 4$$

(2-0) (4-0) 5-1

```
>>> P2([1,1,1,1,1,1])
0
```

```
>>> P2([1,1,0,1,1,1])
2
```

### 문제 3.

0 과 1 로만 이루어진 리스트 A, B 를 입력으로 받는다. 다음 조건을 만족하는 부분 리스트 길이의 최대값을 return 하는 함수를 구현하시오. 그러한 부분 리스트가 없으면 0 을 return 하면 된다.

(A 와 B 는 길이가 1 이상이고, 길이가 서로 같다.)

$$\text{조건: } \text{sum}(A[i:j]) == \text{sum}(B[i:j]), 0 \leq i \leq j \leq \text{len}(A)$$

### 예시

```
>>>A = [0,1,1,0,1,0,1,1,1]
>>>B = [0,0,0,1,0,1,0,1,0]
>>>P3(A, B)
5
```

```
>>>A = [0,0,0,0,0,1]
>>>B = [1,1,1,1,1,0]
>>>P3(A, B)
2
```

```
>>>A = [0,0,0,0,0,1]
>>>B = [1,0,0,0,0,0]
>>>P3(A, B)
6
```

가능한 부분 리스트는 여러가지가 있을 수 있고, 부분리스트의 길이의 최대값을 return 하면 된다.

#### 문제 4.

정수로 이루어진 리스트 `nums` 를 입력으로 받는다. 여기서 몇 개의 수를 뽑는데, 뽑은 수들이 연속적이어야 한다 (순서는 상관없음). 뽑을 수 있는 최대 개수를 `return` 하는 함수를 구현하시오. (리스트의 길이는 1 이상이다.)

#### 예시

```
>>>P4([3, 6, 4, 64, 10, 29, 5, 9, 11])
```

4

설명: 3, 6, 4, 5 를 뽑으면 연속된 4 개의 정수가 된다. 뽑는 방법은 여러가지일 수 있고, 최대 개수만 `return` 하면 된다.

```
>>>P4([-1, 5, 2, -6, 8])
```

1

```
>>>P4([-3, 2, 0, 1, -2, -1])
```

6

{-3, 5

2 =

0

0 : -3, 5

1 : 2

0