

# Computing Foundations for Data Science

## HW 4

**제출기한: 2021/09/23 10:59AM**

### 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 ~ P10.py 를 유지하고, 해당 파일들을 HW4\_학번\_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 Keondo Park 이라면 **HW4\_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.

## 문제 1.

정수(int)로만 구성된 리스트를 변수(argument)로 받고, 그중 두 번 이상 반복되는 정수의 집합(set)을 리턴하는 함수를 작성하시오.

### 예시1)

```
>>> P1([1, 2, 3, 1])  
{1}
```

### 예시2)

```
>>> P1([1, 1, 2, 3, 3, 3])  
{1, 3}
```

### 예시3)

```
>>> P1([1, 2, 3, 4, 5])  
set()
```

## 문제 2.

길이가 같은 리스트 두 개를 변수(arguments)로 받고, 같은 인덱스를 갖는 원소끼리 짝지은 튜플(tuple)의 집합(set)을 리턴하는 함수를 작성하시오.

### <조건>

1. 튜플은 (첫번째 리스트 원소, 두번째 리스트 원소) 순서로 만들어야 함

### 예시1)

```
>>> P2(['red', 'yellow', 'purple'], ['apple', 'banana', 'grape'])  
{('red', 'apple'), ('yellow', 'banana'), ('purple', 'grape')}
```

### 예시2)

```
>>> P2([1, 2, 3], ['computer', 'keyboard', 'mouse'])  
{(1, 'computer'), (2, 'keyboard'), (3, 'mouse')}
```

### 예시3)

```
>>> P2([], [])  
set()
```

### 문제 3.

딕셔너리(dictionary)의 key는 unique함이 보장되지만, value는 그렇지 않다.

하나의 딕셔너리를 변수(argument)로 받고, 그것에 포함된 "distinct values"의 수를 리턴하는 함수를 작성하시오.

#### <조건>

1. 모든 values는 "Hashable" 함

#### 예시1)

```
>>> P3({'red': 1, 'green': 1, 'blue': 2})  
2
```

#### 예시2)

```
>>> P3(((1,2): 'a', 'g': 3, 1: True))  
3
```

#### 예시3)

```
>>> P3(dict())  
0
```

#### 예시4)

```
>>> P3({'a': True, 'b': True, 'c': True})  
1
```

### 문제 4.

일련의 실험을 수행한 후, 특정 다섯 종류의 아원자 입자(subatomic particles)를 탐지할 확률을 {str: float} 딕셔너리 형태로 다음과 같이 저장했다.

```
{'neutron': 0.55, 'proton': 0.21, 'meson': 0.03, 'muon': 0.07, 'neutrino': 0.14}
```

이러한 딕셔너리를 변수(argument)로 받아 탐지 가능성이 가장 큰 입자를 리턴하는 함수를 작성하시오.

#### 예시1)

```
>>> P4({'neutron': 0.55, 'proton': 0.21, 'meson': 0.03, 'muon': 0.07, 'neutrino': 0.14})  
'neutron'
```

#### 예시2)

```
>>> P4({'neutron': 0.11, 'proton': 0.21, 'meson': 0.05, 'muon': 0.09, 'neutrino': 0.12})  
'proton'
```

## 문제 5.

딕셔너리를 변수(argument)로 받아 두 번 이상 나타나는 value를 리스트로 저장하여 리턴하는 함수를 작성하시오.

### 예시1)

```
>>> P5({'red': 1, 'green': 1, 'blue': 2})  
[1]
```

### 예시2)

```
>>> P5({'r': 'a', 'g': 'b', 'b': 'c'})  
[]
```

### 예시3)

```
>>> P5({'a': True, 'b': True, 'c': 2, 'd': 2})  
[True, 2]
```

### 예시4)

```
>>> P5(dict())  
[]
```

## 문제 6.

두 개의 딕셔너리 dct1과 dct2를 변수(arguments)로 받아 dct1과 dct2에 공통으로 포함된 key/value 쌍으로만 만든 새로운 딕셔너리를 리턴하는 함수를 작성하시오.

### 예시1)

```
>>> P6({'a': 1, 'b': True, 'c': [1,2]}, {'a': 1, 'b': 123, 'c': [1,2]})  
{'a': 1, 'c': [1, 2]}
```

### 예시2)

```
>>> P6({'a': 1, 'b': True }, {'c': 1, 'd': 123, 'e': [1,2]})  
{}
```

### 예시3)

```
>>> P6({}, {'c': 1, 'd': 123, 'e': [1,2]})  
{}
```

## 문제 7.

때때로 프로그래머들은 value가 딕셔너리로 이루어진 "Dictionary of dictionaries"를 간단한 데이터베이스로 이용한다.

key가 문자열(string)이고 value가 딕셔너리 형태인 "Dictionary of dictionaries"를 다음과 같이 나타낼 수 있고, 이때 value는 "inner dictionary"라 한다.

```
{ 'jgoodall' : {'surname' : 'Goodall',
               'forename' : 'Jane',
               'born' : 1934,
               'died' : None,
               'notes' : 'primate researcher',
               'author' : ['In the Shadow of Man','The Chimpanzees of Gombe']},
  'franklin' : {'surname' : 'Franklin',
               'forename' : 'Rosalind',
               'born' : 1920,
               'died' : 1957,
               'notes' : 'contributed to discovery of DNA'},
  'rcarson' : {'surname' : 'Carson',
               'forename' : 'Rachel',
               'born' : 1907,
               'died' : 1964,
               'notes' : 'raised awareness of effects of DDT',
               'author' : ['Silent Spring'] }
```

"Dictionary of dictionaries"를 변수(argument)로 받아 "inner dictionaries"가 모두 같은 key 값을 가지면 1, 아니면 0을 리턴하는 함수를 작성하시오.

### <조건>

1. 딕셔너리가 비어있는 경우는 없음
2. 모든 value는 딕셔너리로 되어있음
3. value의 value는 딕셔너리 형태가 아님

### 예시1)

```
>>>P7({'a': {'aa':123, 'ab': [1,2]}, 'b': {'aa': 'bb', 'ab': 'cc'}})
```

1 (Explanation: 모든 value는 같은 key값, {'aa', 'ab'}를 갖는다.)

### 예시2)

```
>>>P7({'A': {1: 'a', 2: 'b'}, 'B': {2: 'c', 3: 'd'}})
```

0

## 문제 8.

다음과 같이 대부분의 원소가 0으로 되어있는 벡터를 "Sparse vector"라 한다.

```
sparse_vector = [1, 0, 0, 0, 0, 0, 3, 0, 0, 0]
```

이때 벡터를 모든 0을 포함한 형태로 리스트에 저장하면 메모리가 낭비되기 때문에 딕셔너리를 이용하여 0이 아닌 원소만 나타내기도 한다. 위의 벡터는 딕셔너리를 사용하여 다음과 같이 나타낼 수 있다. `sparse_vector[0] = 1, sparse_vector[6] = 3` 을 딕셔너리로 표현한 것이다.

```
{0: 1, 6: 3}
```

두 벡터의 합은 같은 위치에 있는 원소들을 원소끼리 더하는 것이다. (element-wise sum)

예를 들어 [1, 2, 3]과 [4, 5, 6]의 합은 [5, 7, 9]이다.

모든 원소가 정수이고, 딕셔너리 형태로 저장된 두 개의 sparse vector를 변수(arguments)로 받아 그들의 합을 새로운 딕셔너리로 저장하여 리턴하는 함수를 작성하시오.

### 예시1)

```
>>> P8({0: 1, 6: 3}, {0: 2, 5: 2, 6: 2, 7: 1})
{0: 3, 5: 2, 6: 5, 7: 1}
```

### 예시2)

```
>>> P8 ({0: 1, 6: 3}, {0: -1, 5: 2, 6: 2, 7: 1})
{6: 5, 5: 2, 7: 1}
```

### 예시3)

```
>>> P8({0: 1, 6: -3}, {0: -1, 6: 3})
{}

```

## 문제 9.

문제 8번에서 설명한 "Sparse vector"를 사용한다.

두 벡터의 내적(dot product)은 대응하는 원소들의 곱을 모두 합하는 것이다.

예를 들어 [1, 2, 3]과 [4, 5, 6]의 내적은  $(1 \times 4) + (2 \times 5) + (3 \times 6) = 32$  이다.

딕셔너리 형태로 저장된 두 개의 sparse vector를 변수(arguments)로 받아 그들의 내적을 계산하여 리턴하는 함수를 작성하시오.

### 예시1)

```
>>> P9({0:1, 6:3}, {0:2, 5:2, 6:2, 7:1})
8

```

### 예시2)

```
>>> P9({0:1, 6:3}, {1:-1, 2:2, 3:2, 4:1})
0

```

## 문제 10.

"Query word" 주어진다면 character중 하나를 꼭 바꿔서 "Changed word"를 만들어야 한다.

단어 집합(set)과 "Query word"를 변수(arguments)로 받아서 "Changed word"가 단어 집합 내의 단어와 하나라도 일치하면 True, 하나도 일치하지 않으면 False를 리턴하는 함수를 작성하시오.

### <조건>

1. 모든 단어는 소문자 알파벳으로만 구성되고 띄어쓰기는 없음
2. "Query word"의 character 단 하나만 바꿀 수 있음

### 예시1)

```
>>> P10({"data", "science"}, "data")
```

False

### 예시2)

```
>>> P10({"data", "science"}, "daaa")
```

True (Explanation: "daaa"의 a 하나를 바꾸면 "data"를 만들 수 있다.)

### 예시3)

```
>>> P10({"data", "science"}, "scienzz")
```

False

### 예시4)

```
>>> P10({"data", "science", "scienze"}, "scienzz")
```

True