

# Computing Foundations for Data Science

## Project 4

**제출기한: 2021/11/02 10:59AM**

- 모든 문제에 해당하는 사항: 뼈대 코드의 함수 이름 및 parameter는 수정하지 말 것. Python Standard Library (collections 등) 사용 가능.
- 주어진 함수 외에 다른 함수를 구현해서 이용해도 된다. 뼈대 코드의 함수(P1 ~ P4)가 return 값만 제대로 전달하면 하면 된다.
- Return 값을 전달하기 위한 용도 외의 불필요한 코드(print 문, 테스트 코드 등) 때문에 채점 진행에 방해가 될 경우 불이익이 있을 수 있다. 제출 전에 불필요한 부분은 지우거나 주석 처리한 후 제출할 것.
- 특히 이번 과제의 경우 graph 탐색 과정을 전부 출력할 경우 채점 진행에 문제가 될 수 있으니 주의 바람. 조교의 컴퓨터에서 실행하는데 10 초 이상 걸릴 경우 틀린 것으로 간주할 예정)

## 문제 1. 방 안에 퍼지는 방귀

M x N 사이즈의 방 안에 뽕뽕이 몇 명이 방귀를 뀌었다. 방귀는 1초가 지나면 인접한 공간 (위, 아래, 왼쪽, 오른쪽)으로 퍼진다. 지독한 방귀이기에 한번 퍼지기 시작하면 사라지지 않는다. 몇 초 만에 방 전체로 방귀가 퍼지는지 시간을 계산하여 return 하는 함수 P1을 구현해야 한다.

- 방에 대한 정보로 M x N 리스트인 *rooms* 를 함수의 파라미터로 받는다.
- M과 N은 1 이상 자연수이다.
- 리스트는 0 또는 1 또는 -1로 되어있다.
  - 0은 방귀가 퍼질 수 있는 빈 공간이다.
  - 1은 처음(0초일 때) 뽕뽕이들이 방귀를 뀌 자리이다. 최소 한 곳 이상에 방귀를 뀌었고, 방귀는 동시에 퍼진다.
  - 방귀는 위, 아래, 왼쪽, 오른쪽으로 퍼질 수 있고, 대각선으로는 퍼질 수 없다.
  - 1초마다 퍼진다.
  - -1은 벽이어서 방귀가 퍼질 수 없다.
- 방 전체, 즉 0인 곳 모두로 방귀가 퍼질 때까지 걸린 시간을 return해야 한다.
  - 방 전체로 방귀가 퍼질 수 없으면 -1을 return한다.
- 처음(0초)부터 방귀가 짝 찬 상태면 0을 return한다.

### 예시1)

```
>>> P1([[[-1, 1],  
          [1, -1]])  
0
```

설명: 처음부터 방귀가 짝 차 있으므로 0을 return

### 예시2)

```
>>> P1([[1, 0]])  
1
```

### 예시3)

```
>>> P1([[0,0,0,0,0,0],  
        [0,0,0,0,0,0],  
        [0,0,0,0,0,0],  
        [0,0,0,0,0,1]])
```

8

### 예시4)

```
>>> P1([[ 0,-1,0,0,0,0],  
        [-1, 0,0,0,0,0],  
        [ 0, 0,0,0,0,0],  
        [ 0, 0,0,0,0,1]])
```

-1

설명: 왼쪽 위 구석은 벽 때문에 방귀가 도달할 수 없으므로 -1을 return

### 예시5)

```
>>> P1([[1,-1,0,0, 0,0],  
        [0,-1,0,0, 0,0],  
        [0, 0,0,0,-1,0],  
        [0, 0,0,0,-1,1]])
```

6

설명: 방귀는 동시에 퍼지기 시작한다.

### 예시6)

```
>>> P1([[-1,1, 0, 0,0],  
        [0,-1,-1,-1,0],  
        [0,-1,-1,-1,0],  
        [0,-1,-1,-1,0],  
        [0, 0, 0, 0,0]])
```

14

## 문제 2. 소셜 미디어

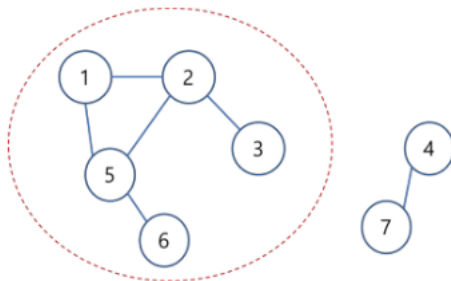
어떤 소셜 미디어에는 1번부터  $n$ 번까지의 회원이 있고, 친구 관계를 tuple로 표현할 수 있다. 예를 들어, (1, 2)라는 것은 1번과 2번이 서로 친구라는 것이다(쌍방향). 즉, 친구 관계는 그래프로 나타낼 수 있다. 이 때, 친구 관계로 이어진 모든 회원을 하나의 클러스터라고 하자. 1번 회원이 속한 클러스터의 회원 수를 return하는 함수 P2를 구현하라.

- 회원 수  $n$  과 친구 관계를 나타내는 tuple의 리스트 *edges* 를 입력으로 받는다.
- $n$ 은 1 이상 자연수이고, edges는 tuple로 이루어진 리스트이다.
  - 리스트 원소의 개수는 0 이상
  - 각 tuple은 서로 다른 2개의 수 (1 이상  $n$  이하)로 이루어져 있다. 똑같은 tuple은 없다.
- 회원 수에는 1번도 포함시킨다.

### 예시1)

```
>>>P2(7, [(1, 2), (2,3), (1,5), (5, 2), (5, 6), (4, 7)])  
5
```

설명: 다음과 같이 1번 회원이 속한 클러스터에는 총 5명의 회원이 있다.



### 예시2)

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

### 예시3)

```
>>>P2(3, [(1,2)])  
2
```

#### 예시4)

```
>>>P2(4, [(1, 2), (2, 1)])
```

```
2
```

#### 참고.

그래프를 표현하는 방법에는 adjacency matrix와 adjacency list를 이용하는 방법이 대표적이다.

(<https://www.geeksforgeeks.org/graph-and-its-representations/>)

### 문제 3. 글자 필터링

필터링된 이미지 속에서 글자(알파벳)가 몇 개인지 알아보는 프로그램을 만들어 보고자 한다. 해당 필터는 이미지에서 글자인 부분은 1, 글자가 아닌 부분은 0으로 바꾼다고 한다. 한 개의 글자는 1이 상, 하, 좌, 우, 대각선으로 인접하여 서로 연결되어 있다. 해당 필터를 적용하여 만든 이미지가 입력으로 주어졌을 때, 입력 이미지 내의 글자의 개수를 return하는 함수 P3를 구현하라.

- 입력 값은  $M \times N$  사이즈의 리스트이며  $M, N$ 은 1 이상 자연수이다.
- 리스트에 숫자 0, 1 외의 값은 없다.
- 원본 이미지 속 글자는 알파벳 대문자로만 구성되어 있다.

#### 예시1)

```
>>> P3[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0],
        [0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0],
        [0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0],
        [0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0],
        [0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0],
        [0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
3
```

설명:

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 1 0 1 1 1 1 1 0
0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0
0 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

⇒ ANT 3글자가 있으므로 3

#### 예시2)

```
>>> P3[[0, 0, 0, 0, 0],
        [0, 1, 1, 0, 0],
        [0, 1, 0, 1, 0],
        [0, 1, 0, 1, 0],
        [0, 1, 1, 0, 0],
        [0, 0, 0, 0, 0]]
1
```

설명:

```
0 0 0 0 0
0 1 1 0 0
0 1 0 1 0
0 1 0 1 0
0 1 1 0 0
0 0 0 0 0
```

⇒ D 한 글자만 존재하므로 1

예시3)

```
>>> P3[[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0],
[0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0],
[0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0],
[0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0],
[0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0],
[0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0],
[0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0],
[0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

4

설명:

```
0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0
0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0
0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0
0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0
0 1 1 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0
0 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 0
0 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0
0 0 1 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0
0 0 1 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

⇒ LOVE 네 글자가 존재하므로 4

#### 문제 4. 보물 찾기

루피가 보물을 찾으려고 한다. 루피의 위치와 보물의 위치는 수직선 위의 점으로 표현된다. 루피는 걷거나 점프를 할 수 있다. 루피의 현재 위치가  $X$ 일 때 걷는다면 1 초 뒤에  $X+1$  또는  $X-1$ 로 갈 수 있고, 점프를 하면 1초 뒤에  $2*X$ 로 갈 수 있다. 남들이 보물을 찾기 전에 빨리 찾아야 한다. 가장 빠르게 찾을 때, 몇 초 만에 출발 지점부터 보물이 있는 곳에 도달할 수 있을지 걸린 시간을 return하는 함수를 구현하라.

- 루피의 현재 위치  $L$  과 보물의 위치  $T$  를 입력 받는다.
- $0 \leq L \leq 100000$ ,  $0 \leq T \leq 100000$ ,  $L$ 과  $T$ 는 정수

##### 예시1)

```
>>>P4(5, 17)
4
```

설명:  $5 \rightarrow 10 \rightarrow 9 \rightarrow 18 \rightarrow 17$ , 이렇게 가면 4초 만에 갈 수 있다. 도달하는 방법은 여러 가지가 있을 수 있다.

##### 예시2)

```
>>>P4(8, 1)
7
```

설명:  $8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$

##### 예시3)

```
>>>P4(100, 100)
0
```

설명 : 움직일 필요가 없다.

##### 예시4)

```
>>>P4(123, 12345)
36
```



## 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 PROJ4\_학번\_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 박건도라면 **PROJ4\_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
- Return 값을 전달하기 위한 용도 외의 불필요한 코드(print 문, 테스트 코드 등) 때문에 채점 진행에 방해가 될 경우 불이익이 있을 수 있음. 제출 전에 불필요한 부분은 지우거나 주석 처리한 후 제출.
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있음.
- 늦은 제출은 받지 않음.
- 표절 검사를 수행하여 발각될 경우 성적 F 부여.