

Computing for Data Science

Project #5

제출 기한: 2021.04.21 오전 10:59

- 모든 문제에 해당하는 사항: 뼈대 코드의 함수 이름 및 parameter 는 수정하지 말 것. Python Standard Libray(collections 등) 사용 가능.

- 주어진 함수 외에 다른 함수를 구현해서 이용해도 된다. 뼈대 코드의 함수가 제대로 return 하면 된다.

P1.

$M \times N$ 으로 된 방안에 뽕뽕이 몇 명이 방귀를 끼었다. 방귀는 1 초가 지나면 인접한 공간 (위, 아래, 왼쪽, 오른쪽)으로 퍼진다. 지독한 방귀여서 한번 퍼지면 사라지지 않는다. 몇 초 만에 방 전체로 방귀가 퍼지는지를 계산해서 return 하는 함수를 구현해야 한다.

방에 대한 정보 $M \times N$ 리스트인 `rooms` 를 함수의 입력으로 받는다.

- M 과 N 은 1 이상 자연수이다.
- 리스트는 0 또는 1 또는 -1 로 되어있다.
 - 0 은 방귀가 퍼질 수 있는 빈 공간이다.
 - 1 은 처음(0 초일 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.
 - ◆ 1 초마다 퍼진다.
 - -1 은 벽이어서 방귀가 퍼질 수 없다.
- 방 전체 즉, 0 인 곳 모두로 방귀가 퍼질 때까지 걸린 시간을 return 해야 한다.
 - 방 전체로 방귀가 퍼질 수 없으면 -1 을 return 한다.
 - 처음(0 초)부터 방귀가 짝 찬 상태면 0 을 return 한다.

예시.

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

설명: 처음부터 방귀가 짝 차있으므로 0 을 return

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

```
>>>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
```

```
>>>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 -1 을 return

```
>>>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 퍼진다.

```
>>>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
```

P2.

Facebook 에는 1 번부터 n 번 까지의 회원이 있고, 친구 관계를 tuple 로 표현할 수 있다. 예를 들어, (1, 2)라는 것은 1 번과 2 번이 서로 친구라는 것이다(쌍방향). 즉, 친구 관계는 그래프로 나타낼 수 있다. 이 때, 친구 관계로 이어진 모든 회원을 하나의 클러스터라고 하자. **1 번** 회원이 속한 클러스터의 회원 수를 return 하는 함수를 구현하시오.

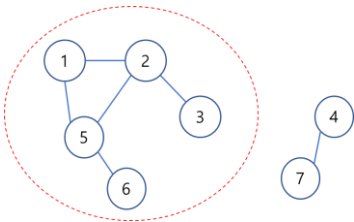
회원 수 n 과 친구 관계를 나타내는 tuple 의 리스트 edges 를 입력으로 받는다.

- n 은 1 이상 자연수이고, edges 는 tuple 로 이루어진 리스트이다.
 - 리스트 원소의 개수는 0 이상
 - 각 tuple 은 서로 다른 2 개의 수(1 이상 n 이하)로 이루어져 있다. 똑같은 tuple 은 없다.
- 회원 수에는 1 번도 포함시킨다.

예시.

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

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



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

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

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

참고.

그래프를 표현하는 방법에는 adjacency matrix 와 adjacency list 를 이용하는 방법이 대표적이다.

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

P3.

M x N 리스트인 `world` 를 함수의 입력으로 받는다. 리스트의 원소는 1 또는 0 으로 이루어져 있고, 1 은 땅, 0 은 물을 의미한다. 이 때, 인접한 땅으로 연결되어 있고, 물로 둘러 싸여 있는 지역을 섬이라고 하자. 섬의 개수를 return 하는 함수를 구현하시오.

- M 과 N 은 1 이상 자연수이다.
- 리스트 바깥은 물이라고 가정한다.
- 인접해 있다는 것은 상, 하, 좌, 우 네 방향 중 한 곳에서 붙어 있는 것이고, 대각선 방향은 인접해 있는 것이 아니다.
- 섬이 아닌 땅은 없다.

예시.

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

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

설명: 대각선 방향은 인접해 있는 것이 아니다.

```
>>>P3([[1,1,1,1,1],
        [1,1,1,1,1],
        [1,1,1,1,1],
        [1,1,1,1,1]])
1
```

```
>>>P3([[0,0,0,0,0],
        [0,0,0,0,0],
        [0,0,0,0,0],
        [0,0,0,0,0]])
0
```

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

4

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

2

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

2

P4.

루피가 보물을 찾으려 한다. 루피의 위치와 보물의 위치는 수직선 위의 점으로 표현된다. 루피는 걸거나 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 1 초 뒤에 $2*x$ 로 갈 수 있다. 남들이 보물을 찾기 전에 빨리 찾아야 한다. 가장 빠르게 찾을 때, 몇 초 만에 보물이 있는 곳에 도달할 수 있는 지 걸린 시간을 return 하는 함수를 구현하시오.

- 함수의 입력으로 루피의 현재 위치 L 과 보물의 위치 T 를 입력받는다.
 - $0 \leq L \leq 100000, 0 \leq T \leq 100000, L$ 과 T 는 정수

예시.

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

설명: 5 -> 10 -> 9 -> 18 -> 17 이렇게 가면 4 초만에 갈 수 있다. 도달하는 방법은 여러가지가 있을 수 있다.

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

설명: 8 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1

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

설명: 움직일 필요가 없다.

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

주의사항

- 코드를 Jupyter Notebook 에서 작성하였더라도 python 파일(.py)로 제출할 것.
- 함수가 의도한 값을 Return 하는지를 확인. (print 와 혼동하지 말 것)
- 파일명은 P1.py ~ P4.py 를 유지하고, 해당파일들을 PROJ5_학번_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 Keondo Park 이라면 **PROJ5_2020_12345_KeondoPark.zip** 으로 압축하여 제출.
- 예시로 제시한 입력값 외에 조교가 랜덤으로 생성한 입력값으로 코드가 잘 작성되었는지 테스트할 것이다.
- 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.