Image Processing

실습 8.

2021. 04. 25.

실습 수업 소개

- **과목 홈페이지**
  - 충남대학교 사이버 캠퍼스 (http://e-learn.cnu.ac.kr)

- **TA 연락처**
  - 신준호
  - wnsgh578@naver.com

- **튜터 연락처**
  - 한승오
  - so.h4ns@gmail.com

- **실습 중 질문사항**
  - 실시간 수업중 질문 or 메일을 통한 질문
  - 메일로 질문할 때 [IP] 를 제목에 붙여주세요

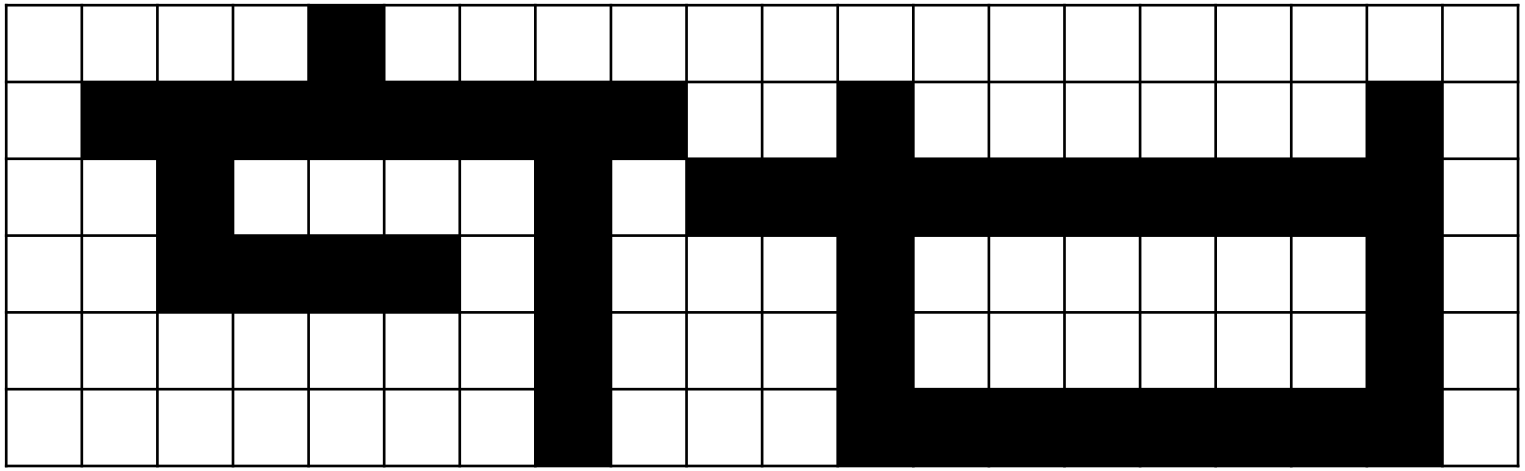
실습 수업 소개

- **실습 출석**
  - 사이버캠퍼스를 통해 Zoom 출석
  - Zoom 퇴장 전 채팅 기록[학번 이름] 남기고 퇴장
  - 위 두 기록을 통해 출석 체크 진행 예정

목 차

Connectedness

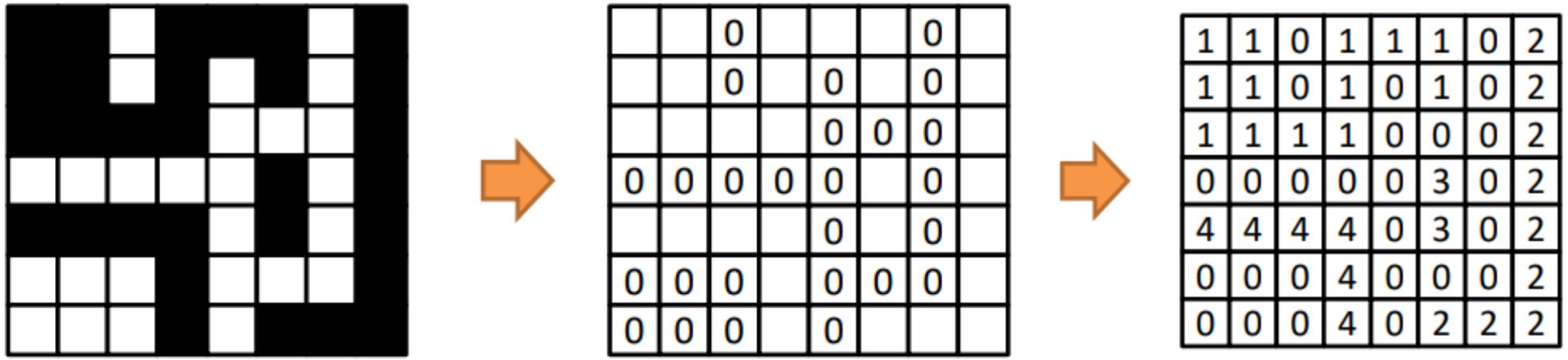
- **4-neighborhood vs 8-neighborhood**



4-neighborhood : Two sets of connected components
8-neighborhood : One set of connected components

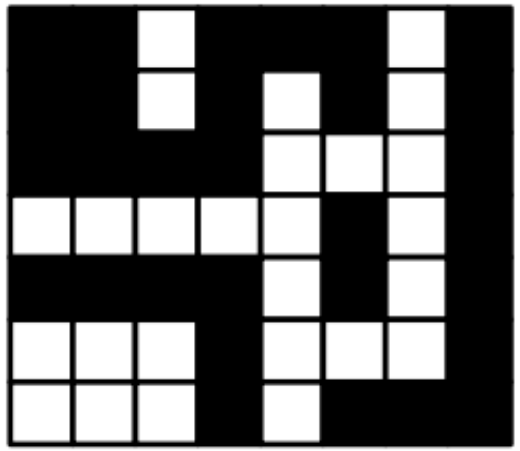
Connectedness

- **Connected components labeling**



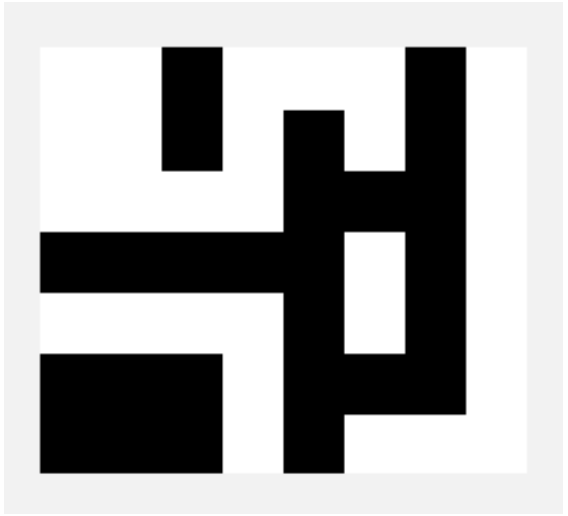
Connectedness

- **Connected components labeling**



B

zero : white
one : black

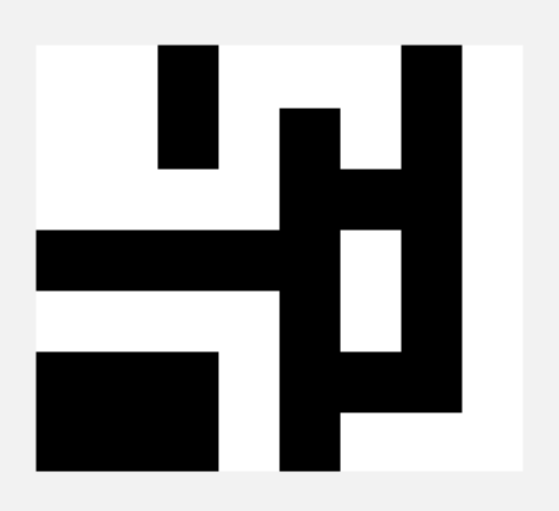


B

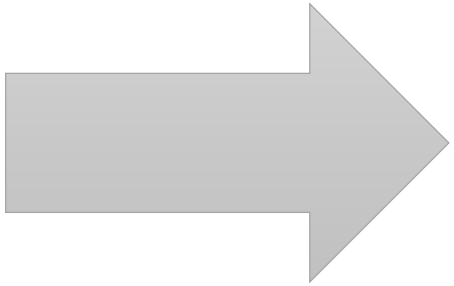
zero : black
one : white

Connectedness

- **Connected components labeling**



B



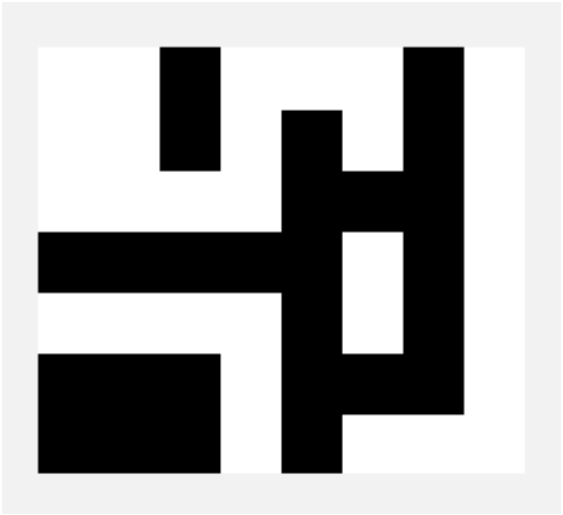
1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	3	0	2
4	4	4	4	0	3	0	2
0	0	0	4	0	0	0	2
0	0	0	4	0	2	2	2

L

zero : black
one : white

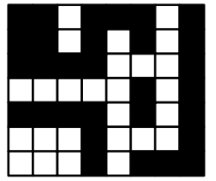
Connectedness

- **Connected components labeling**
  - B 이미지 만들기



B

zero : black
one : white



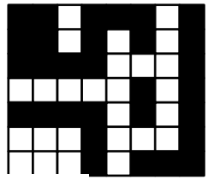
B

```
def main():
    B = np.array(
        [[0, 0, 1, 0, 0, 0, 1, 0],
         [0, 0, 1, 0, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 1, 1, 0],
         [1, 1, 1, 1, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 0, 1, 0],
         [1, 1, 1, 0, 1, 1, 1, 0],
         [1, 1, 1, 0, 1, 0, 0, 0]])
    B = 1 - B
    B = (B*255).astype(np.uint8)
    cv2.imwrite('binary_image.png',B)
```

0 -> 1
1 -> 0

Connectedness

- **Connected components labeling**
  - 0 채우기(이미 채워진 상태)



B

```
def main():
    B = np.array(
        [[0, 0, 1, 0, 0, 0, 1, 0],
         [0, 0, 1, 0, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 1, 1, 0],
         [1, 1, 1, 1, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 0, 1, 0],
         [1, 1, 1, 0, 1, 1, 1, 0],
         [1, 1, 1, 0, 1, 0, 0, 0]])
    B = 1 - B
    B = (B*255).astype(np.uint8)
    cv2.imwrite('binary_image.png',B)
```

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	0	0		0			

L

zero : black
one : white

Connectedness

- **Connected components labeling**
  - 빈 곳은 -1로 채우기

```
[[-1 -1  0 -1 -1 -1  0 -1]
 [-1 -1  0 -1  0 -1  0 -1]
 [-1 -1 -1 -1  0  0  0 -1]
 [ 0  0  0  0  0 -1  0 -1]
 [-1 -1 -1 -1  0 -1  0 -1]
 [ 0  0  0 -1  0  0  0 -1]
 [ 0  0  0 -1  0 -1 -1 -1]]
```

zero : black
one : white

```
def main():
    B = np.array(
        [[0, 0, 1, 0, 0, 0, 1, 0],
         [0, 0, 1, 0, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 1, 1, 0],
         [1, 1, 1, 1, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 0, 1, 0],
         [1, 1, 1, 0, 1, 1, 1, 0],
         [1, 1, 1, 0, 1, 0, 0, 0]])
    B = 1 - B
    #B = (B*255).astype(np.uint8)
    #cv2.imwrite('binary_image.png',B)

    B = B * -1
    print(B)
```

Connectedness

- **Connected components labeling**
  - labeling 하기

```
[[1 1 0 1 1 1 0 2]
 [1 1 0 1 0 1 0 2]
 [1 1 1 1 0 0 0 2]
 [0 0 0 0 0 3 0 2]
 [4 4 4 4 0 3 0 2]
 [0 0 0 4 0 0 0 2]
 [0 0 0 4 0 2 2 2]]
```

L

zero : black
one : white

```
def main():
    B = np.array(
        [[0, 0, 1, 0, 0, 0, 1, 0],
         [0, 0, 1, 0, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 1, 1, 0],
         [1, 1, 1, 1, 1, 0, 1, 0],
         [0, 0, 0, 0, 1, 0, 1, 0],
         [1, 1, 1, 0, 1, 1, 1, 0],
         [1, 1, 1, 0, 1, 0, 0, 0]])
    B = 1 - B
    #B = (B*255).astype(np.uint8)
    #cv2.imwrite('binary_image.png',B)

    B = B * -1
    #print(B)

    B = labeling(B)
    print(B)
```

Connectedness

- **Connected components labeling**
  - labeling 하기

```
[[1 1 0 1 1 1 0 2]
 [1 1 0 1 0 1 0 2]
 [1 1 1 1 0 0 0 2]
 [0 0 0 0 0 3 0 2]
 [4 4 4 4 0 3 0 2]
 [0 0 0 4 0 0 0 2]
 [0 0 0 4 0 2 2 2]]
```

zero : black
one : white

```python
# 실습에서는 4-neighbor
def labeling(B, neighbor=4):
    label = 0

    h, w = B.shape
    dst = B.copy()

    for row in range(h):
        for col in range(w):
            if dst[row, col] == -1:
                coordinates = find_set_pos(dst, row, col, [])
                label += 1

                while coordinates:
                    r, c = coordinates.pop()
                    dst[r, c] = label

    return dst
```

Connectedness

- **Connected components labeling**
  - labeling 하기

```
[[1 1 0 1 1 1 0 2]
 [1 1 0 1 0 1 0 2]
 [1 1 1 1 0 0 0 2]
 [0 0 0 0 0 3 0 2]
 [4 4 4 4 0 3 0 2]
 [0 0 0 4 0 0 0 2]
 [0 0 0 4 0 2 2 2]]
```

L

zero : black
one : white

```python
def find_set_pos(src, row, col, coordinates):
    if (row, col) in coordinates or src[row, col] != -1:
        return coordinates

    coordinates.append((row, col))
    h, w = src.shape
    # up
    if row > 0:
        coordinates = find_set_pos(src, row-1, col, coordinates)
    # down
    if row < h-1:
        coordinates = find_set_pos(src, row+1, col, coordinates)
    # left
    if col > 0:
        coordinates = find_set_pos(src, row, col-1, coordinates)
    # right
    if col < w-1:
        coordinates = find_set_pos(src, row, col + 1, coordinates)

    # 중복 제거
    return list(set(coordinates))
```

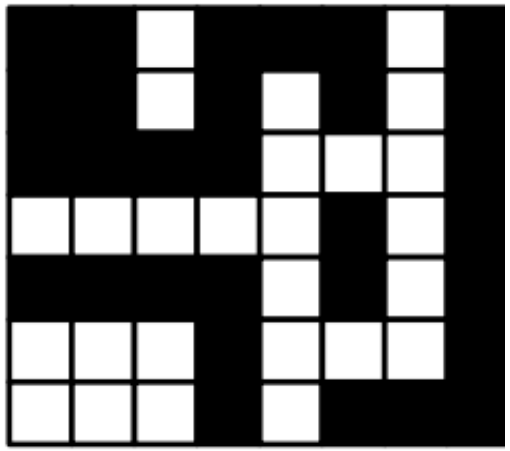
Connectedness

- **Connected components labeling**
  - labeling 하기

```
[[1 1 0 1 1 1 0 2]
 [1 1 0 1 0 1 0 2]
 [1 1 1 1 0 0 0 2]
 [0 0 0 0 0 3 0 2]
 [4 4 4 4 0 3 0 2]
 [0 0 0 4 0 0 0 2]
 [0 0 0 4 0 2 2 2]]
```

L

zero : black
one : white



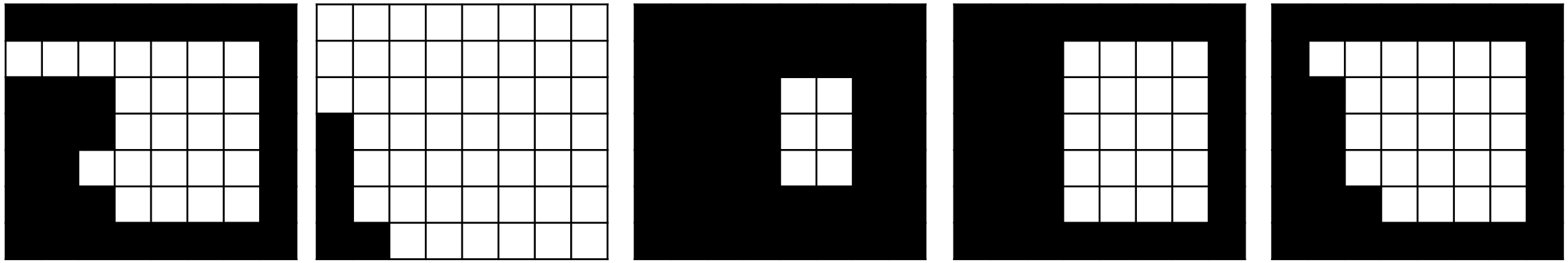
B

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	3	0	2
4	4	4	4	0	3	0	2
0	0	0	4	0	0	0	2
0	0	0	4	0	2	2	2

L

과제

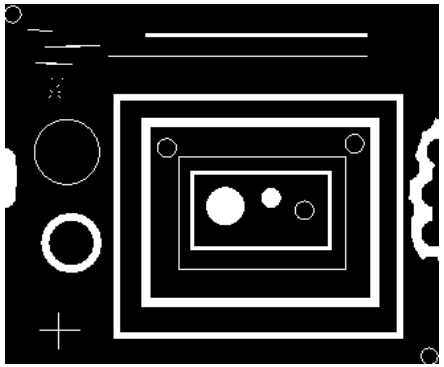
- **Morphological Operations**
  - Dilation, Erosion, Opening, Closing



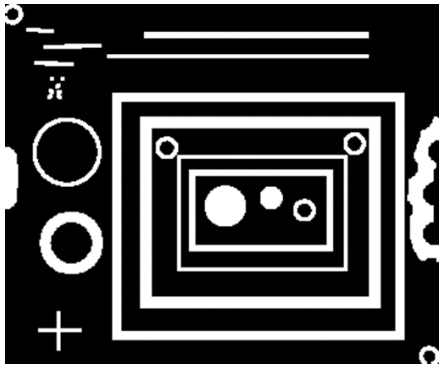
Original image | Dilation | Erosion | Opening | Closing

과제

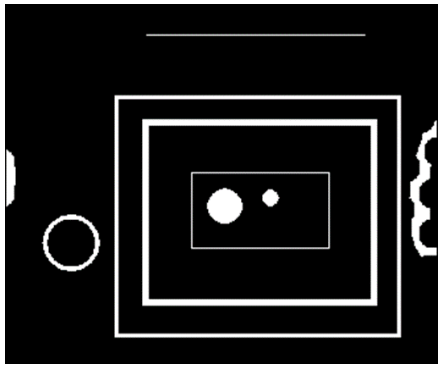
- **Morphological Operations**
  - Dilation, Erosion, Opening, Closing



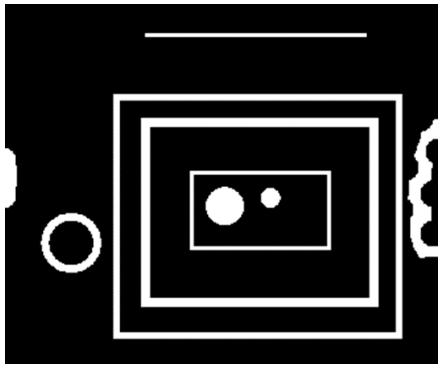
Original image



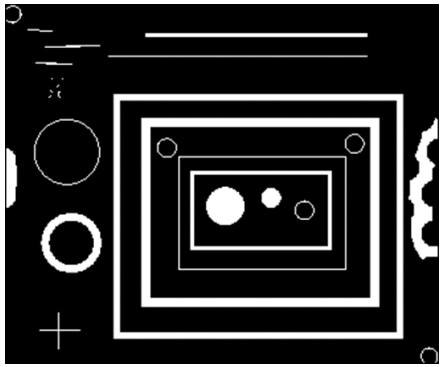
Dilation



Erosion



Opening



Closing

과제

- **morphology.py**
  - 4개의 morphology operation 함수를 구현
    - dilation
    - erosion
    - opening
    - closing

  - 보고서에 결과 5개의 이미지를 포함하여 작성
    - Original, dilation, erosion, opening, closing
    - 두 번째 페이지의 이미지를 사용한 결과만 포함
    - main에 시각화 코드가 주어져 있음

과제

- **제출 방법**
  - 코드 파일
    - 구현 결과가 포함된 python 파일(.py)
  - 보고서
    - [IP]201900000_홍길동_2주차_과제.pdf
    - 보고서 양식 사용
    - PDF 파일 형식으로 제출(pdf가 아닌 다른 양식으로 제출시 감점)
  - 제출 파일
    - [IP]201900000_홍길동_2주차_과제.zip
    - .py 파일과 pdf 보고서를 하나의 파일로 압축한 후, 양식에 맞는 이름으로 제출

출석체크

- **Zoom 퇴장 전, [학번 이름]을 채팅창에 올린 후 퇴장해 주시기 바랍니다.**

QnA