



# 알고리즘 11주차

## Weighted graph2

MMC 연구실

박사 과정 문희찬

## 조교 소개

- 문희찬
- 컴퓨터공학과 대학원 석사과정
- MMC연구실 (A1406)
- HCMoon@hallym.ac.kr



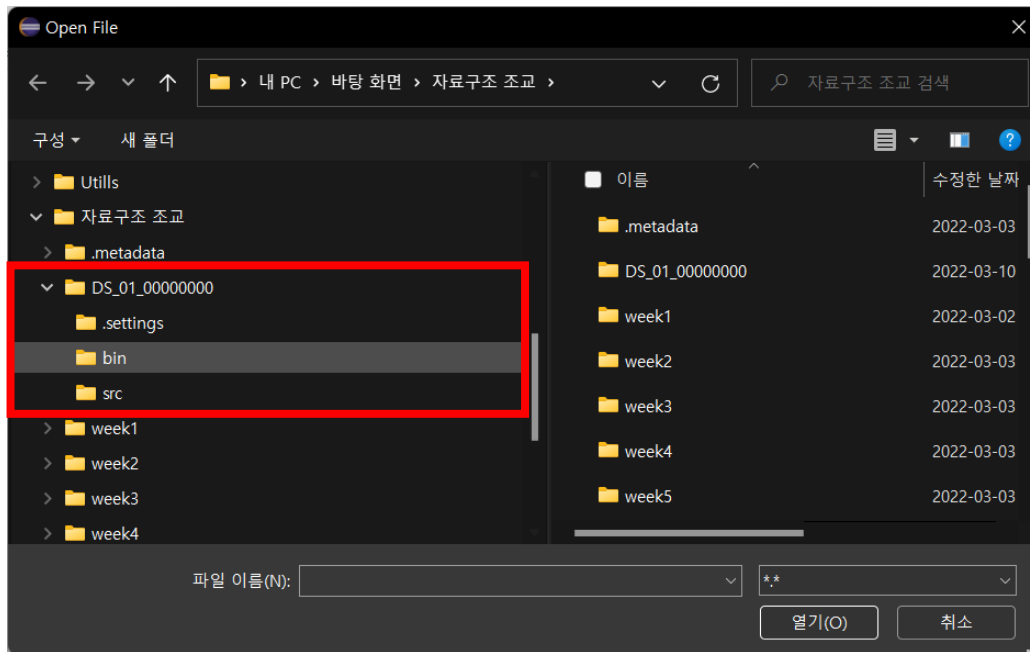
## 실습 수업 진행 방식

- 쉬는 시간 없이 1시간 30분 수업 (화장실 자유롭게 다녀오세요)
- 출석체크 : 수업 시작, 수업 끝날 때 체크
- 수업 시작 30분 뒤부터, 확인 문제를 해결한 학생은 검사 받고 퇴실
- 과제 진행 중 모르는 부분은 메일로 질문

## 과제 설명

- 알고리즘 수업은 Eclipse를 사용하여 코드를 작성합니다.
- 확인 문제 및 과제를 전부 해결하여 제출해주세요.
- 과제 제출 시 **프로젝트 폴더를 압축**해서 제출합니다.
- 과제의 채점은 프로젝트의 실행 결과를 기준으로 점수를 매깁니다.
- 컨닝 금지, 모르는 것이 있으면 저에게 질문해주세요.  
(메일 주소 확인)

# 과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

- 프로젝트이름 : AL\_(주차)\_(학번)

예) AL\_11\_00000000

- \*.java파일만 제출하면 안됩니다.

- 제출양식을 반드시 지켜주세요!

# 확인문제

Package Name : wgraph  
Class Name : WGraph

```
public int[] shortestPath(int v) {  
    boolean s[] = new boolean[n];  
    int dist[] = new int[n];  
    int u;  
    // 이곳에 코드 작성  
    return dist;  
}  
  
public int[] negativePath(int v) {  
    int dist[] = new int[n];  
    // 이곳에 코드 작성  
    return dist;  
}  
  
public int[][] allShortestPath() {  
    int[][] distance = weight;  
    int[][] distanceP;  
    int k, i, j;  
    // 이곳에 코드 작성  
    return distance;  
}
```



## 확인문제

Package Name : wgraph  
Class Name : WGraphTest

<terminated> WGraphTest (1) [Java Application]

```
0 -> 0: 0  
0 -> 1: 2  
0 -> 2: 4  
0 -> 3: 5  
0 -> 4: 3
```

```
0 -> 0: 0  
0 -> 1: 0  
0 -> 2: 2  
0 -> 3: 5  
0 -> 4: -1  
0 -> 5: 2
```

```
0 5 4 - - -  
- 0 2 7 - -  
- - 0 6 11 -  
- - - 0 3 8  
- - - - 0 8  
- - - - - 0  
0 5 4 10 13 18  
- 0 2 7 10 15  
- - 0 6 9 14  
- - - 0 3 8  
- - - - 0 8
```

# 실습 과제

## 1. transitive closure matrix





# 과제

Package Name : graph

Class Name : Graph

```
public boolean[][] transitive() {  
    boolean[][] distance = arc;  
    boolean[][] distanceP;  
    int k, i, j;  
  
    //  
  
    return distance;  
}
```



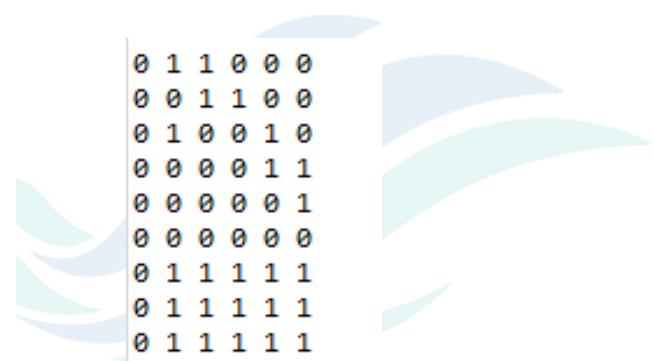
# 확인문제

Package Name : graph  
Class Name : GraphTest

```
Graph gr1 = new Graph(6);
gr1.insertEdge(0, 1);
gr1.insertEdge(0, 2);
gr1.insertEdge(1, 2);
gr1.insertEdge(1, 3);
gr1.insertEdge(2, 1);
gr1.insertEdge(2, 4);
gr1.insertEdge(3, 4);
gr1.insertEdge(3, 5);
gr1.insertEdge(4, 5);

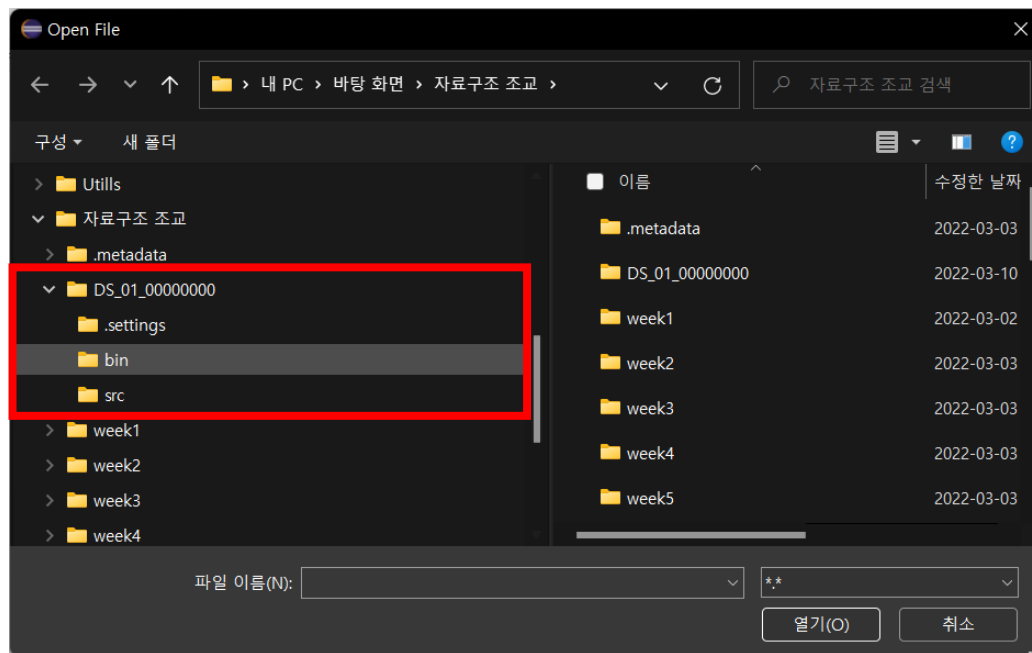
for(int i = 0; i < 6; i++) {
    for(int j = 0; j < 6; j++)
        if(gr1.arc[i][j])
            System.out.print("1 ");
        else
            System.out.print("0 ");
    System.out.println();
}

boolean[][] isPath = gr1.transitive();
for(int i = 0; i < 6; i++) {
    for(int j = 0; j < 6; j++)
        if(isPath[i][j])
            System.out.print("1 ");
        else
            System.out.print("0 ");
    System.out.println();
}
```



0	1	1	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	0	0	1	1
0	0	0	0	0	1
0	0	0	0	0	0
0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	0	0	0	1	1
0	0	0	0	0	1
0	0	0	0	0	0

# 과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

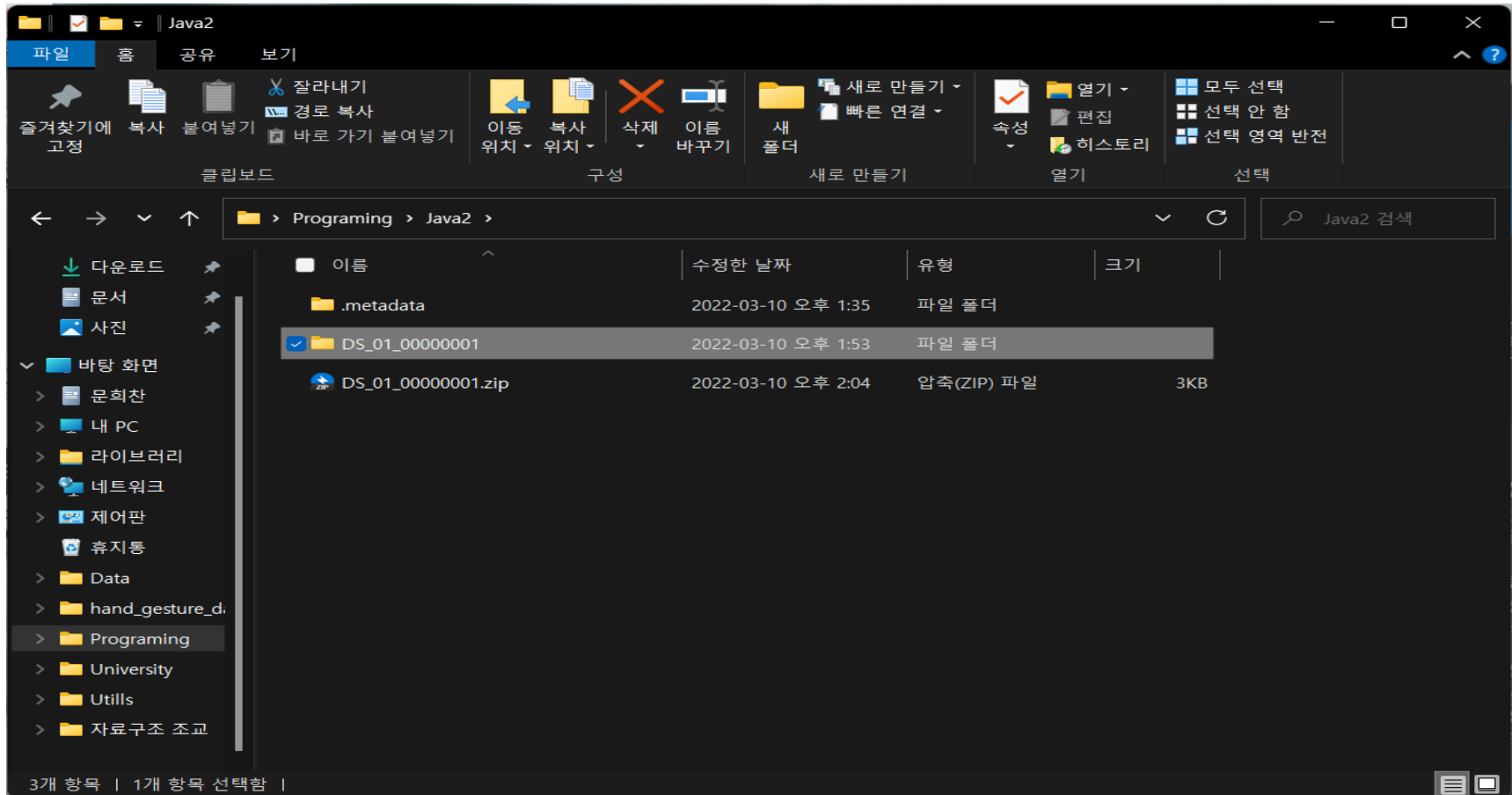
- 프로젝트이름 : AL\_(주차)\_(학번)

예) AL\_11\_00000000

- \*.java파일만 제출하면 안됩니다.

**제출양식을 반드시 지켜주세요**

# 과제 제출 방법



- 반드시 **프로젝트 폴더를 압축**하여 제출