

On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

서명: 강동인 학번: 21500002

Antenna Result

```
// antenna.txt
8
9
0 1
0 5
1 2
1 6
2 3
2 6
3 6
4 5
6 7
```



Result

```
vertex[0..7] =    0    1    2    3    4    5    6    7
color[0..7] =    0    0    0    0    0    0    0    0
DFS0[0..7] =    0    5    4    1    6    7    3    2
CCID[0..7] =    1    1    1    1    1    1    1    1
DFS parent[0..7] = -1    0    3    6    5    0    1    6
BFS0[0..7] =    0    5    1    4    6    2    7    3
DistTo[0..7] =    0    1    2    3    2    1    2    3
BFS parent[0..7] = -1    0    1    6    5    0    1    6
```

DFS sequence

```
*1st call: dfs(0)    0번 노드 방문, visit[0] == false, 방문으로 표기 후 순서에 추가
*2nd call: dfs(5)    5번 노드 방문, visit[5] == false, 방문으로 표기 후 순서에 추가
*3rd call: dfs(4)    4번 노드 방문, visit[4] == false, 방문으로 표기 후 순서에 추가
4th call: dfs(5)    5번 노드 방문
    check 5    visit[5] == true, 건너뛸
    4 done    4번 노드의 모든 인접노드 탐색 완료

5th call: dfs(0)    0번 노드 방문
```

```

        check 0    visit[0] == true, 건너뛴
        5 done     5번 노드의 모든 인접노드 탐색 완료

*6th call: dfs(1)  1번 노드 방문, visit[1] == false, 방문으로 표기 후 순서에 추가
*7th call: dfs(6)  6번 노드 방문, visit[6] == false, 방문으로 표기 후 순서에 추가
*8th call: dfs(7)  7번 노드 방문, visit[7] == false, 방문으로 표기 후 순서에 추가
9th call: dfs(6)   6번 노드 방문
        check 6    visit[6] == true, 건너뛴
        7 done     7번 노드의 모든 인접노드 탐색 완료

*10th call: dfs(3) 3번 노드 방문, visit[3] = false, 방문으로 표기 후 순서에 추가
*11th call: dfs(2) 2번 노드 방문, visit[2] = false, 방문으로 표기 후 순서에 추가
12th call: dfs(1)  1번 노드 방문
        check 1    visit[1] == true, 건너뛴
13th call: dfs(3)  3번 노드 방문
        check 3    visit[3] == true, 건너뛴
14th call: dfs(6)  3번 노드 방문
        check 6    visit[3] == true, 건너뛴
        2 done     2번 노드의 모든 인접노드 탐색 완료

15th call: dfs(6)  6번 노드 방문
        check 6    visit[6] == true, 건너뛴
        3 done     3번 노드의 모든 인접노드 탐색 완료

16th call: dfs(2)  2번 노드 방문
        check 2    visit[2] == true, 건너뛴
17th call: dfs(1)  1번 노드 방문
        check 1    visit[1] == true, 건너뛴
        6 done     2번 노드의 모든 인접노드 탐색 완료

18th call: dfs(2)  2번 노드 방문
        check 2    visit[2] == true, 건너뛴
        dfs(0)     5번 노드 방문
        check 0    visit[0] == true, 건너뛴
        1 done     1번 노드의 모든 인접노드 탐색 완료

        0 done     0번 노드의 모든 인접노드 탐색 완료

```

DFS call 앞에 *가 붙어있으면 순서를 표기한 DFS call 입니다. 따라서 DFS의 sequence는

: 0 -> 5 -> 4 -> 1 -> 6 -> 7 -> 3 -> 2

BFS sequence

i	Queue	adj_list[i]	visit[currentNode]
*0		5, 1	true (preprocess)
*5	1	0, 4	false
*1	4	0, 6, 2	false
*4	6, 2	5	false
*6	2	7, 3, 2, 1	false
*2	7, 3, 2	6, 3, 1	false
*7	3, 2, 3	6	false
*3	2, 3	6, 2, 1	false
2	3	6, 3, 1	true
3		6, 2	true

- 방문하지 않은 노드만 큐에 넣어줍니다.
- Queue 항목의 붉은 색으로 표시된 숫자는 새로 추가된 노드들입니다.
- adj_list[i] 항목의 붉은 색으로 표시된 숫자는 큐에 추가될 노드들입니다.
- visit[currentNode]가 true이라면 이미 방문한 노드이기 때문에 프로세스를 건너뜁니다.