

1. 구현 아이디어

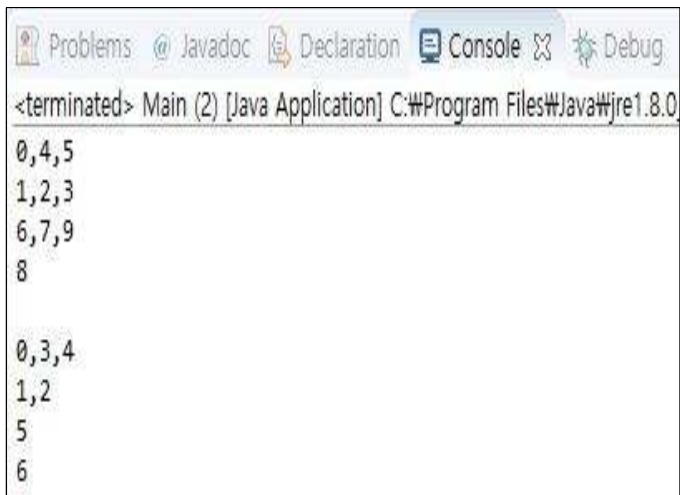
주어진 graph의 data를 가지고 adacency list로 구현하였다. 이후 DFS 알고리즘을 사용하여 connected component를 구하였다. 이때 DFS 알고리즘은 stack을 활용하였다. 출력은 java library의 treeset을 사용하였다.

2. 구현

```
DFS(source)
1 Stack S ← ∅
2 TreeSet T ← ∅
3 PUSH(S,source)
4 while S ≠ ∅
5     do v ← POP(S)
6     if visited[v] ≠ true
7         ADD(T,v)
8         visited[v] = true
9         for each dest ∈ Adj[v]
10            do if visited[dest] ≠ true
11                PUSH(S, dest)
12 if T ≠ ∅
13     for each e ∈ T
14         print(e)
```

4~11번 줄에서 기존 recursive로 구현한 DFS 알고리즘을 stack을 사용하여 구현하였다. 12~14번 줄에서는 출력하기 위한 부분이다.

3. 결과



```
<terminated> Main (2) [Java Application] C:\Program Files\Java\jre1.8.0
0,4,5
1,2,3
6,7,9
8

0,3,4
1,2
5
6
```