<알고리즘및실습> - 과제 4

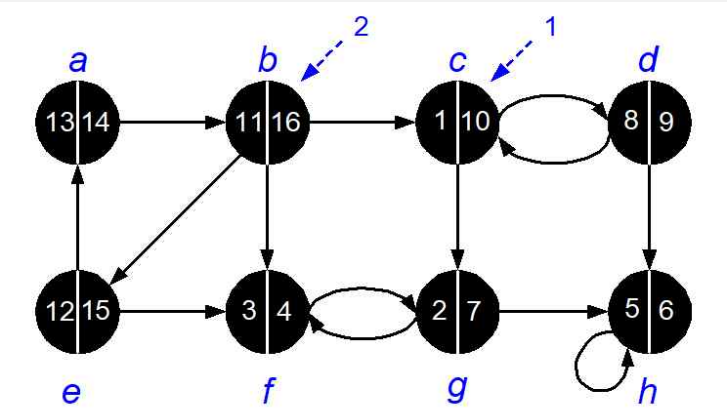
[문제 1] 강 연결 요소(SCC: Strongly Connected Component) 찾기

방향 그래프에서 각 정점에서 다른 모든 정점으로 도달할 수 있는 최대 부 그래프를 강 연결 요소(SCC)라 한다. 아래의 설명을 참고하여 방향 그래프의 SCC를 찾는 프로그램을 구현하여라.

주어진 방향 그래프에서 SCC를 찾는 방법은 다음과 같다.

1) 임의의 한 정점으로부터 시작하여 Visit Time과 Finishing Time을 표기하면서 DFS 실행

- Visit Time: DFS 호출로 해당 정점을 처음 방문한 시각 (정점의 왼쪽)
- Finishing Time: DFS 재귀 호출이 반환되어 해당 정점을 떠나는 시각 (정점의 오른쪽)



2) Finishing Time을 기준으로 정점(Node) 내림차순 정렬

- 위 그래프의 경우 b, e, a, c, d, g, h, f 의 순서로 정점들이 정렬됨.

3) 주어진 방향 그래프의 모든 간선(Edge)의 방향을 반대로 바꾼 그래프 생성

4) 앞선 2)의 과정에서 정렬된 정점의 순서로 3)의 그래프에서 DFS 반복 실행

5) 앞선 4)의 과정에서 한 번의 DFS로 방문할 수 있는 정점들이 하나의 SCC 구성

- 하나의 방향 그래프는 그 구조에 따라 여러 개의 SCC를 가질 수 있음.

SCC를 찾는 프로그램에서 입력으로 주어지는 방향 그래프는 다음 조건을 만족한다.

- **n (1 ≤ n ≤ 100) 개의 정점**과 **m (1 ≤ m ≤ 1,000) 개의 방향 간선**으로 구성됨.
- 모든 정점은 **1 ~ n** 사이의 정수로 번호가 할당되며, 각 정점에 할당된 번호는 모두 다름.
- 모든 간선은 **방향 간선**이고, 한 정점에서 임의의 다른 정점으로 가는 경로가 **존재하지 않을 수도 있음**. (입력 예시 1 참고)

이제 방향 그래프에서 SCC를 찾는 프로그램은 다음의 입력과 출력을 가진다.

[입력]

- 첫 줄에는 정점의 개수 **n**, 간선의 개수 **m**, 초기 DFS 순회 시작 정점 번호 **s**가 주어진다.
- 이후, **m**개의 줄에 걸쳐 한 줄에 하나씩 간선의 정보가 주어진다.
  방향 그래프이므로 간선의 출발 정점과 도착 정점의 순서는 구별된다. **(u, v) ≠ (v, u)**
  중복 입력되는 방향 간선은 존재하지 않으나, 두 정점이 양방향으로 연결될 수는 있다.

[출력]

- **더 작은 번호의 정점을 가지는 SCC부터 한 줄씩 정점 번호가 오름차순으로 출력된다.**
  예를 들어, 두 개의 SCC [3, 4, 2]와 [5, 1]을 가지는 방향 그래프는 프로그램 수행 결과로
  1, 5
  2, 3, 4
  가 콘솔에 출력된다. (가장 작은 정점 번호 1을 가진 SCC부터 우선 출력됨.)

입력 예시 1

```
3 2 1   ↦ n = 3, m = 2, s = 1
2 3     ↦ 정점 2에서 정점 3으로 가는 방향 간선
3 2     ↦ 정점 3에서 정점 2로 가는 방향 간선
```

출력 예시 1

```
1     ↦ 첫 번째 SCC 정점 출력
2 3   ↦ 두 번째 SCC 정점 출력
```

입력 예시 2

```
5 7 1   ↦ n = 5, m = 7, s = 1
1 2     ↦ 정점 1에서 정점 2로 가는 방향 간선
1 4     ↦ 정점 1에서 정점 4로 가는 방향 간선
5 1     ↦ 정점 5에서 정점 1로 가는 방향 간선
3 5     ↦ 정점 3에서 정점 5로 가는 방향 간선
4 3     ↦ 정점 4에서 정점 3으로 가는 방향 간선
3 1     ↦ 정점 3에서 정점 1로 가는 방향 간선
2 3     ↦ 정점 2에서 정점 3으로 가는 방향 간선
```

출력 예시 2

```
1 2 3 4 5   ↦ SCC 정점 출력
```

입력 예시 3

```
8 14 3   ↦ n = 8, m = 14, s = 3
1 2
5 1
5 6
2 5
2 6
6 7
7 6
2 3
3 7
3 4
4 3
7 8
4 8
8 8
```

출력 예시 3

```
1 2 5  ↦ 첫 번째 SCC 정점 출력
3 4    ↦ 두 번째 SCC 정점 출력
6 7    ↦ 세 번째 SCC 정점 출력
8      ↦ 네 번째 SCC 정점 출력
```

<주의사항>

- 소스코드에는 적절한 주석을 붙여야 하고,

- 코드 제일 앞부분에 주석으로 자신의 학번, 이름, 사용한 자료구조(인접리스트/행렬, 연결리스트/배열등)을 서술.

- 다른 학생의 소스코드를 표절하는 경우에는 0점 처리되므로 스스로 구현할 것.

- 과제 제출 기한: 12/9(월) 밤 12시