

알고리즘 HW3

컴퓨터공학부 최덕경

2016-10399

- Summary of this assignment

어떤 방향이 있는 그래프에 속한 모든 정점의 쌍 (u, v) 에 대해서 u 에서 v 로, v 에서 u 로 도달 가능할 때 이를 "Strongly Connected" 그래프라고 한다. 그리고 한 방향 그래프가 주어졌을 때 그 그래프에 속한 Strongly Connected Subgraph 중 가장 큰 것을 Strongly Connected Component라고 한다. 이번 과제에서는 한 방향 그래프가 주어졌을 때 모든 SCC를 구하는 알고리즘을 구현해볼 것이다. 단 한 번은 인접 리스트로, 다른 한 번은 인접 행렬로 그래프를 표현하여 각각 알고리즘을 구현해볼 것이다. 각각의 그래프 표현 방식은 한 정점의 이웃 정점을 방문하는 시간 복잡도가 다르기 때문에 이와 관련된 대부분의 알고리즘도 그래프의 표현 방식에 따라 시간 복잡도가 달라지게 된다. 이를 검증하기 위해, 이번 과제에서 각각의 방식으로 알고리즘을 구현해보고, 그 실행시간의 차이를 직접 비교 분석해볼 것이다.

- Development environment

(1) 사용 언어 : Python 3.6.7

(2) 실험 환경 : Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-45-generic x86_64)

(3) 실험 방법 : python3 main.py

- Running time of my algorithms on four input test data

<main.py> 코드의 runtime 함수는 해당 함수를 해당 인자로 호출하고 그 실행시간을 측정한다.

	인접 리스트 기반 알고리즘 (초)	인접 행렬 기반 알고리즘 (초)
in1.txt	0.0017857644706964493	0.01953496504575014
in2.txt	0.005971070379018784	0.06342738494277
in3.txt	0.008579152636229992	0.1346438778564334
in4.txt	0.010210799984633923	0.2497715801000595

```
Runtime of algorithm based on adjacency list (sec) : 0.0017857644706964493
Runtime of algorithm based on adjacency array (sec) : 0.01953496504575014
```

```
Runtime of algorithm based on adjacency list (sec) : 0.005971070379018784
Runtime of algorithm based on adjacency array (sec) : 0.06342738494277
```

```
Runtime of algorithm based on adjacency list (sec) : 0.008579152636229992
Runtime of algorithm based on adjacency array (sec) : 0.1346438778564334
```

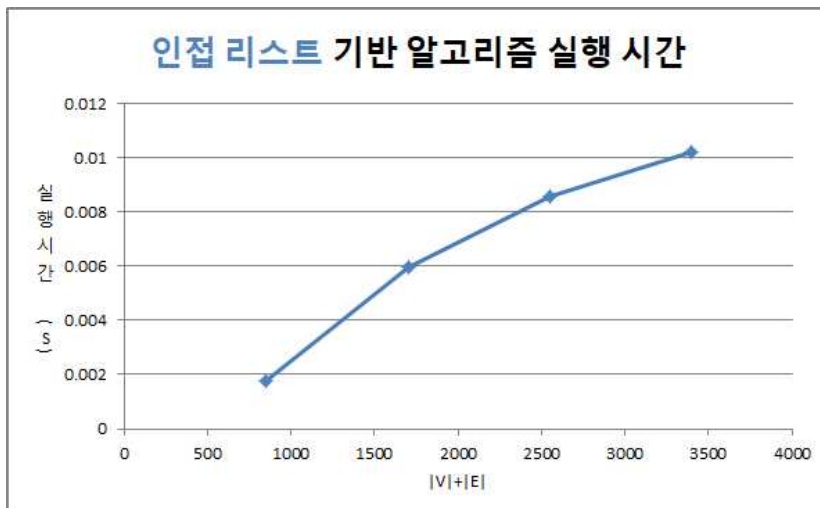
```
Runtime of algorithm based on adjacency list (sec) : 0.010210799984633923
Runtime of algorithm based on adjacency array (sec) : 0.2497715801000595
```

▶ 차례대로 in1.txt, in2.txt, in3.txt, in4.txt에 대하여 두 알고리즘의 실행 시간을 측정한 결과

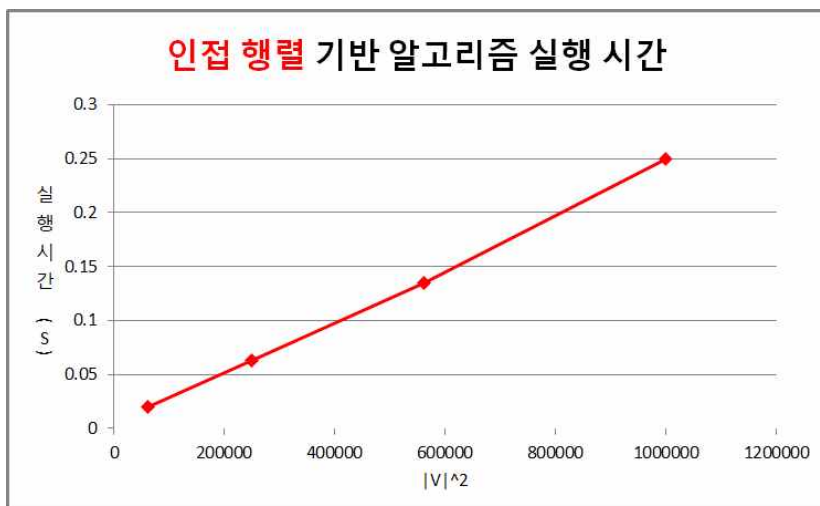
- Analyze the result (Check if my algorithms show expected time complexity)

	$ V $ = 정점의 개수	$ E $ = 간선의 개수	$ V + E $	$ V ^2$
in1.txt	250	600	850	62500
in2.txt	500	1200	1700	250000
in3.txt	750	1800	2550	562500
in4.txt	1000	2400	3400	1000000

▶ 입력 데이터 크기



▶ 인접 리스트 기반 알고리즘 실행 시간 : 인접 리스트로 구현하면 DFS는 $O(|V|+|E|)$ 의 시간이 걸리기 때문에 DFS에 기반한 이 알고리즘의 시간 복잡도도 $O(|V|+|E|)$ 라고 예상하고 그래프를 그렸더니 예상과 똑같이 $|V|+|E|$ 에 대하여 실행시간이 거의 선형적임을 볼 수 있었다.



▶ 인접 행렬 기반 알고리즘 실행 시간 : 인접 행렬로 구현하면 DFS는 $O(|V|^2)$ 의 시간이 걸리기 때문에 DFS에 기반한 이 알고리즘의 시간 복잡도도 $O(|V|^2)$ 라고 예상하고 그래프를 그렸더니 예상과 똑같이 $|V|^2$ 에 대하여 실행시간이 거의 선형적임을 볼 수 있었다.

∴ 두 알고리즘은 예상 시간 복잡도를 그대로 가지면서 올바르게 구현되었다.