

[511643] 자료구조**실습 #11 보고서**

이름	곽영주
학번	20175105
소속 학과/대학	빅데이터
분반	03 (담당교수: 김태운)

<주의사항>

- 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- **각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.**
 - 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
- 스마트캠퍼스 제출 데드라인: **2020.05.27. ~ 2020.06.02.(화요일) 23:59 // 1 주**
 - 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)
 - 주말/휴일/학교행사 등 모든 날짜 카운트 함
 - 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함
 - 예외 없음
- 스마트캠퍼스에 아래의 파일을 제출 해 주세요
 - 보고서(**PDF 파일로 변환 후 제출**)
 - 보고서 파일명에 이름과 학번을 입력 해 주세요.
 - 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (미 제출시 감점)

<개요>

이번 과제는 그래프 및 탐색 알고리즘을 구현하는 내용입니다.

<실습 과제>

[Q 0] 요약 [배점: 10]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.
답변: 이번 과제를 완성하기 위해 강의노트가 좋은 참고서가 되었습니다. 그리고 강의를 복습하면서 11 주차 강의 그래프 내용까지 잘 이해할 수 있었습니다.

[Q 1] 인접리스트를 이용한 그래프 표현 [10 점]

강의자료 p.12, p.13 을 참고하여 p.11 에 표시된 그래프를 인접리스트로 구현하세요.
다음으로, `public void print()` 메소드를 구현하여 그래프를 저장하고있는 인접리스트를 출력하세요. 출력 결과는 아래와 같아야 합니다.

<터미널 출력 예시>

인접리스트를 이용한 그래프:

[0] => 1 => 2

[1] => 0 => 2 => 3

[2] => 0 => 1 => 3

[3] => 1 => 2

* 필요시, Edge 클래스를 일부 수정해서 사용해도 됩니다.

* 출력 결과는 위의 출력 예시와 완벽히 동일해야 합니다(즉, Edge 가 출력되는 순서도 동일해야 함)

터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q1] 인접리스트를 이용한 그래프 표현.txt

<terminated> Q1Test [Ja

인접리스트를 이용한 그래프:

[0] => 1 => 2

[1] => 0 => 2 => 3

[2] => 0 => 1 => 3

[3] => 1 => 2

[Q 2] 깊이우선탐색(DFS) [20 점]

강의자료를 참고하여 DFS 클래스를 구현하세요. 강의자료 p.19 와 같이 그래프(즉, 주어진 그래프를 구현한 인접리스트)를 구성하고, 깊이우선탐색을 실행하세요. 터미널 출력 결과가 p.19 의 결과와 동일해야 합니다.

터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q2] 깊이우선탐색(DFS).txt

<terminated> Q2Test [Jav

DFS 방문 순서:

0 2 3 9 8 1 4 5 7 6

[Q 3] 너비우선탐색(BFS) [20 점]

강의자료를 참고하여 BFS 클래스를 구현하세요. 강의자료 p.26 과 같이 그래프(즉, 주어진 그래프를 구현한 인접리스트)를 구성하고, 깊이우선탐색을 실행하세요. 터미널 출력 결과가 p.26 의 결과와 동일해야 합니다.

터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q3] 너비우선탐색(BFS).txt

```
<terminated> Q3Test [Jav  
BFS(0) 방문 순서:  
0 2 1 3 9 8 4 5 7 6
```

[Q 4] 위상정렬 [20 점]

강의자료를 참고하여 TopologicalSort 클래스를 구현하세요. 강의자료 p.33 과 같이 그래프를 구성하고, 위상정렬을 실행하세요. 터미널 출력 결과의 예시는 p.33 을 참고하세요. 출력 결과가 p.33 의 결과와 같아도 되고, 또는 결과가 다르더라도 위상정렬 순서에 맞기만 하면 됩니다.

터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q4] 위상정렬.txt

```
<terminated> Q4Test [Java Applicat  
위상 정렬:  
[2, 0, 1, 4, 5, 3, 6, 7, 8]
```

[Q 5] Kosaraju 알고리즘을 이용한 강연결성분 찾기 [20 점]

강의자료를 참고하여 Kosaraju 클래스를 구현하세요. 강의자료 p.41 과 같이 그래프 G 를 구성하고, 강연결성분을 모두 찾아 터미널에 출력하세요. 터미널 출력 결과의 예시는 아래와 같습니다.

<터미널 출력 예시, Ex 1>	<터미널 출력 예시, Ex 2>
강 연결 성분: [2,8,9], [0,6,3], [5], [1,4,7]	강 연결 성분: [1] 2,9,8 [2] 0,6,3 [3] 5 [4] 1,4,7

* 참고: 어떤 강연결성분이 먼저 출력되는지, 그리고 각각의 강 연결성분에서 어떤 정점이 먼저 출력되는지는 위의 예시와 달라도 됩니다. 즉, [5], [8,9,2], [7,1,4], [3,6,0]으로 출력되어도 됩니다. (정답만 맞으면 OK). 출력 형식은 예시와 달라도 됩니다. 하지만 발견한 강연결성분은 정확해야 합니다.

터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q5] Kosaraju 알고리즘을 이용한 강연결성분 찾기.txt

```
<terminated> Q5Test [Java Application] C:\W
[2, 9, 8] [0, 6, 3] [5] [1, 4, 7]
```

끝! 수고하셨습니다 ☺