

실습 10 – 그래프의 기본 연산

- 실습 목표
 - 인접 리스트 방식의 그래프를 생성할 수 있다.
 - 그래프의 기본 연산들(DFS, BFS, 신장 트리 등)을 구현할 수 있다.

인접 리스트를 이용한 그래프의 생성

- **(10.1.)** 노드 수와 에지 정보가 입력되면, 이를 이용하여 인접 리스트 형태로 무방향성 그래프를 구성

입력	인접 리스트를 이용한 그래프의 구성
6	<pre> graph LR 0 --> 2 1 --> 0 2 --> 3 2 --> 5 2 --> 0 3 --> 2 3 --> 4 4 --> 3 4 --> 5 5 --> 2 5 --> 4 </pre>
0 2	
0 1	
2 5	
5 4	
3 4	
3 4	
2 3	

- struct node {int data; struct node *next;}
- graph = (struct node *)calloc(노드 수, sizeof(struct node *))
- 에지 (0, 2)에 대해 graph[0]에 노드 2를 추가하고, graph[2]에 노드 0를 추가
- 노드 순서는 중요하지 않으므로, insert_front 방식으로 추가

그래프의 기본 연산

- Degree가 가장 큰 노드의 degree와 그 노드에 연결된 노드들의 리스트를 출력
 - 앞의 예에서 노드 2의 degree가 3으로 가장 크며, $3 \rightarrow 5 \rightarrow 0$ 의 순서로 리스트를 출력

(10.2.)

- 프로그램 6.1의 DFS를 구현하고, main 함수에서 호출
- 프로그램 6.2의 BFS를 구현하고, main 함수에서 호출

(10.3.)

- 프로그램 6.1의 DFS를 수정하여, DFS 신장트리를 출력
 - 앞의 예에서 DFS 신장 트리:
 $(0,1) (0,2) (2,3) (3,4) (4,5)$