

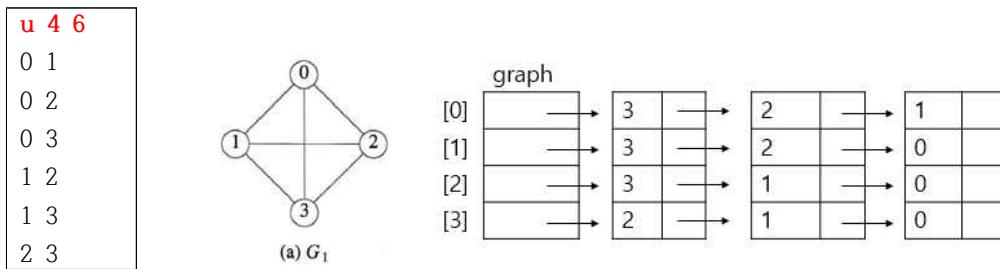
자료구조응용

17. Adjacency List, Adjacency Multilists

1. 다음과 같이 파일 입력을 통해 무방향 그래프(undirected graph)나 방향 그래프(directed graph)를 인접리스트(adjacency-list)로 구성하는 프로그램을 작성하시오.

(1) 입력파일(input.txt) 및 자료구조

- * 입력파일의 첫 줄은 그래프 종류 (u : undirected graph, d : directed graph), 정점 (vertex) 수와 간선(edge)의 수를 나타냄
- * 정점을 나타내는 숫자는 0부터 시작됨



(2) 실행순서

- ① 그래프 종류, 정점, 간선의 수를 파일에서 입력받음
- ② 그래프 종류에 따라 간선을 하나씩 입력받으면서 인접리스트를 구성함.
 - * 입력 되는 노드는 항상 헤더노드가 가리키는 처음 노드로 입력되게 함(항상 리스트의 앞에 노드를 삽입)
- ③ 각 정점에 대해 인접하고 있는 정점들을 (vertices adjacent from a vertex v)을 출력하시오.

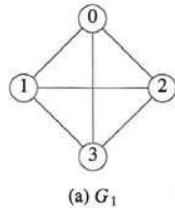
(3) 실행 예

```
C:\WINDOWS\system32\cmd.exe
<<<<<<< Adjacency List >>>>>>>>
adjList[0] : 3 2 1
adjList[1] : 3 2 0
adjList[2] : 3 1 0
adjList[3] : 2 1 0
계속하려면 아무 키나 누르십시오 . . .
```

2. 다음과 같이 사용자로부터 정보를 입력받아서 무방향그래프(undirected graph)를 Adjacency multilist로 구성하여 각 정점에 부속(속해있는)되는 간선을 출력하는 프로그램을 작성하라.

(1) 입력파일(input.txt) 및 자료구조

4 6
0 1
0 2
0 3
1 2
1 3
2 3



adjLists	
[0]	N0 [] 0 1 N1 N3
[1]	N1 [] 0 2 N2 N3
[2]	N2 [] 0 3 0 N4
[3]	N3 [] 1 2 N4 N5
	edge (0,1)
	edge (0,2)
	edge (0,3)
	N4 [] 1 3 0 N5
	edge (1,2)
	edge (1,3)
	N5 [] 2 3 0 0
	edge (2,3)

The lists are

- vertex 0: $N0 \rightarrow N1 \rightarrow N2$
- vertex 1: $N0 \rightarrow N3 \rightarrow N4$
- vertex 2: $N1 \rightarrow N3 \rightarrow N5$
- vertex 3: $N2 \rightarrow N4 \rightarrow N5$

※ 입력파일의 첫 줄은 정점(vertex) 수와 간선(edge)의 수를 나타냄

※ 그래프의 정점은 0부터 시작됨

(2) 구현 세부사항

- ① 하나의 간선 (i, j)을 표현하는 방법

Adjacency List	$\text{adjList}[i] \xrightarrow{\square} \dots \boxed{j} \xrightarrow{\square} \dots$ $\text{adjList}[j] \xrightarrow{\square} \dots \boxed{i} \xrightarrow{\square} \dots$	두 개의 노드
Adjacency Multilists	$\text{adjMulList}[i] \xrightarrow{\square} \dots \xrightarrow{\text{정점 } i \text{만 따라가자!}} \boxed{i} \xrightarrow{\square} \dots$ $\text{adjMulList}[j] \xrightarrow{\square} \dots \xrightarrow{\text{정점 } j \text{만 따라가자!}} \boxed{j} \xrightarrow{\square} \dots$	한 개의 간선 노드를 두 개의 연결리스트가 공유함

(3) 실행순서

- ① 정점과 간선의 수를 입력받음

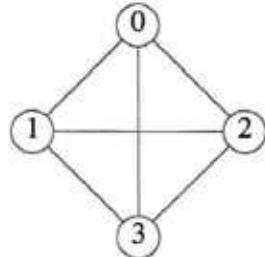
- ② 그래프를 구성하는 간선을 하나씩 입력받으면서 adjacency multilist를 구성함

※ 같은 간선이 두 번 입력되지 않음을 가정함

- ③ 각 정점에 대해 부속된 간선(edges incident to a vertex v)을 출력하기

(입력데이터 순서대로 출력, 헤더노드 정점이 먼저 오게 출력)

(4) 입력 파일 및 실행 예 (* 그래프 G1에 대한 두 가지 입력의 경우)



(a) G_1

① case 1

input1 - 메모장

```

파일(F) 편집(E) 서식(O) 보기(V)
도움말(H)

4 6
0 1
0 2
0 3
1 2
1 3
2 3

```

C:\WINDOWS\system32\cmd.exe

```

<<<<<< edges incident to each vertex >>>>>>
간선의 정점 출력 순서 - 입력데이터 순서대로
edges incident to vertex 0 : (0, 1) (0, 2) (0, 3)
edges incident to vertex 1 : (0, 1) (1, 2) (1, 3)
edges incident to vertex 2 : (0, 2) (1, 2) (2, 3)
edges incident to vertex 3 : (0, 3) (1, 3) (2, 3)

간선의 정점 출력 순서 - 헤더노드 정점이 먼저 오기
edges incident to vertex 0 : (0, 1) (0, 2) (0, 3)
edges incident to vertex 1 : (1, 0) (1, 2) (1, 3)
edges incident to vertex 2 : (2, 0) (2, 1) (2, 3)
edges incident to vertex 3 : (3, 0) (3, 1) (3, 2)
계속하려면 아무 키나 누르십시오 . . .

```

② case 2

input2 - 메모장

```

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

4 6
2 3
0 1
0 3
1 2
0 2
1 3

```

C:\WINDOWS\system32\cmd.exe

```

<<<<<< edges incident to each vertex >>>>>>
간선의 정점 출력 순서 - 입력데이터 순서대로
edges incident to vertex 0 : (0, 1) (0, 3) (0, 2)
edges incident to vertex 1 : (0, 1) (1, 2) (1, 3)
edges incident to vertex 2 : (2, 3) (1, 2) (0, 2)
edges incident to vertex 3 : (2, 3) (0, 3) (1, 3)

간선의 정점 출력 순서 - 헤더노드 정점이 먼저 오기
edges incident to vertex 0 : (0, 1) (0, 3) (0, 2)
edges incident to vertex 1 : (1, 0) (1, 2) (1, 3)
edges incident to vertex 2 : (2, 3) (2, 1) (2, 0)
edges incident to vertex 3 : (3, 2) (3, 0) (3, 1)
계속하려면 아무 키나 누르십시오 . . .

```

■ 제출 형식

- 솔루션 이름 : DS_17
- 프로젝트 이름 : 1, 2
- 솔루션 폴더를 압축하여 제출할 것.
- 학습관리시스템에 과제를 올릴 때 제목:
1차 제출: 학번_이름_DS_17(1), 2차 제출: 학번_이름_DS_17(2)
제출은 2회 걸쳐 가능(수정 시간 기준으로 처리)