

자료구조응용

15. Adjacency List, Adjacency Multilists

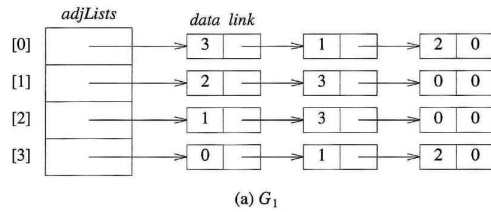
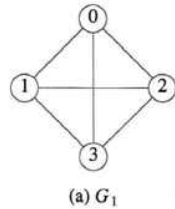
1. 다음과 같이 파일 입력을 통해 무방향 그래프(undirected graph)나 방향 그래프(directed graph)를 인접리스트(adjacency-list)로 구성하는 프로그램을 작성하시오.

(1) 입력파일(input.txt) 및 자료구조

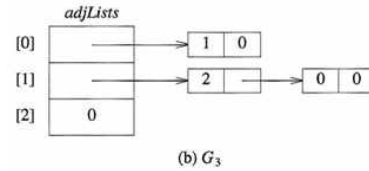
※ 입력파일의 첫 줄은 그래프 종류 (u : undirected graph, d : directed graph), 정점 (vertex) 수와 간선(edge)의 수를 나타냄

※ 정점을 나타내는 숫자는 0부터 시작됨

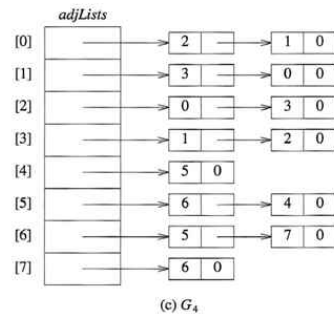
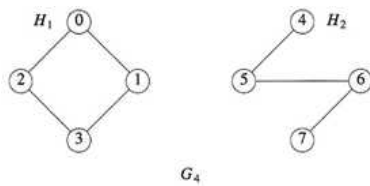
u 4 6
0 1
0 2
0 3
1 2
1 3
2 3



d 3 3
0 1
1 2
1 0



u 8 7
0 1
0 2
1 3
2 3
4 5
5 6
6 7



(2) 실행순서

- ① 그래프 종류, 정점, 간선의 수를 입력받음
- ② 그래프 종류에 따라 간선을 하나씩 입력받으면서 인접리스트를 구성함
 - ※ 항상 헤더노드가 가리키는 처음 노드로 입력되게 함
- ③ 각 정점에 대해 인접하고 있는 정점들을 (vertices adjacent from a vertex v)을 출력하기

(3) 실행 예 (G1, G3, G4)

```
C:\WINDOWS\system32\cmd.exe
```

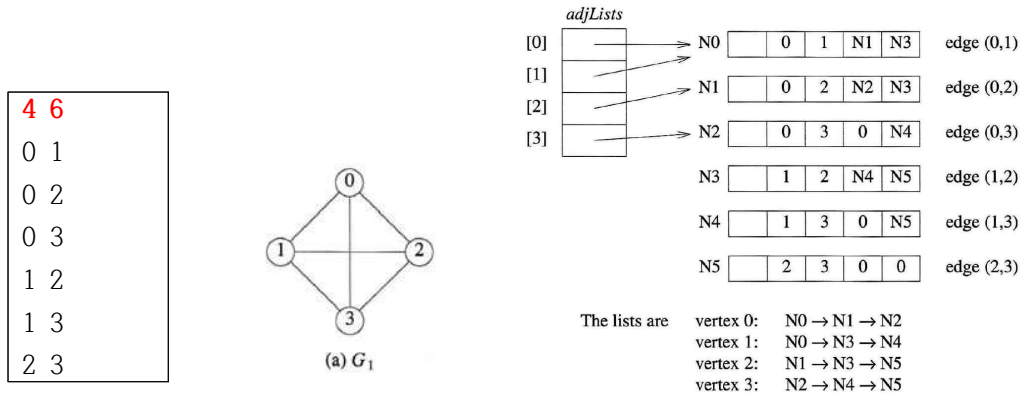
```
<<<<<<<<< Adjacency List >>>>>>>>>>>>  
adjList[0] :   3   2   1  
adjList[1] :   3   2   0  
adjList[2] :   3   1   0  
adjList[3] :   2   1   0  
계속하려면 아무 키나 누르십시오 . . .
```

[illegible]

```
C:\WINDOWS\system32\cmd.exe
<<<<<<<<< Adjacency List >>>>>>>>>>>>
adjList[0] : 2      1
adjList[1] : 3      0
adjList[2] : 3      0
adjList[3] : 2      1
adjList[4] : 5
adjList[5] : 6      4
adjList[6] : 7      5
adjList[7] : 6
계속하려면 아무 키나 누르십시오 . . .
```

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

(1) 입력파일(input.txt) 및 자료구조



- ※ 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	<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">adjList[i]</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">j</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">i</div> <div style="margin: 0 5px;">→</div> <div>...</div> </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">adjList[j]</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">i</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">j</div> <div style="margin: 0 5px;">→</div> <div>...</div> </div>	두 개의 노드
Adjacency Multilists	<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">adjMulList[i]</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">j</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</div> </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">adjMulList[j]</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">i</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">j</div> <div style="margin: 0 5px;">→</div> <div style="border: 1px solid black; padding: 2px 5px; margin-right: 5px;">...</div> </div>	한 개의 간선 노드를 두 개의 연결리스트가 공유함

② Adjacency Multilists 에 노드 추가방법

- 입력된 (i, j)의 간선 노드를 생성 ※ createEdge()
- adjMulList[i]를 따라 가면서 마지막 노드로 추가한 후
- adjMulList[j]를 따라 가면서 마찬가지로 마지막 노드로 연결함 ※ addEdge()

③ 노드 타입 및 함수원형

```
typedef enum { FALSE, TRUE } tBoolean;
typedef struct edge* tEdgePointer;
typedef struct edge
{
    tBoolean marked; // TRUE or FALSE
    int vertex1;
    int vertex2;
    tEdgePointer link1;
    tEdgePointer link2;
}tEdge;
tEdgePointer* adjMulList; // * 동적할당으로 구현

tEdgePointer createEdge(int ver1, int ver2); // create the edge node
void addEdge(tEdgePointer temp, int ver1, int ver2);
void printMultilist(int numVertex, int choice); // choice: 0 or 1
```

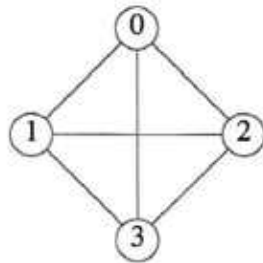
(3) 실행순서

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 세 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 오게
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 오게
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 . . .
```

✓ TA/Tutor는 adjacency list가 아닌 adjacency multilist로 제대로 구현하였는지 소스레벨에서 확인하기 바랍니다.

■ 제출 형식

- 공학인증 시스템(ABEEK)에 과제를 올릴 때 제목:
- 1차 제출: 학번 이름 DS-15(1), 2차 제출: 학번 이름 DS-15(2)
- 솔루션 이름 : DS-15
- 프로젝트 이름 : 1, 2
- 실행화면을 캡처하여 한글파일에 추가 후 솔루션 폴더에 포함.
- 한글 파일명 : 학번_이름_실습결과.hwp
- 솔루션 폴더를 압축하여 게시판에 제출할 것.
- 압축 파일 명: 학번_이름_DS-15.zip