

자료구조응용

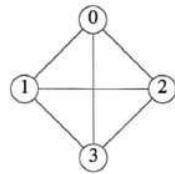
4. Adjacency List, Adjacency Multilists

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

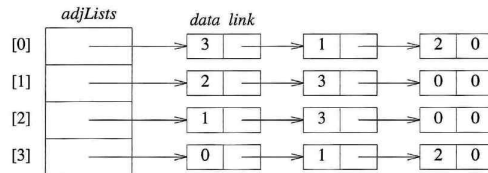
(1) 입력파일(input.txt) 및 자료구조

- ※ 입력파일의 첫 줄은 그래프 종류 (u : undirected graph, d : directed graph), 정점 (vertex) 수와 간선(edge)의 수를 나타냄
- ※ 정점을 나타내는 숫자는 0부터 시작됨
- ※ 주의: 교재의 adjacency list 그림이 본 문제의 데이터 입력 순서에 따른 결과는 아님

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



(a) G_1

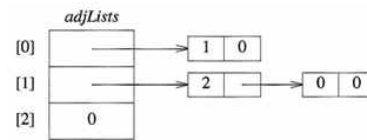


(a) G_1

d 3 3
0 1
1 2
1 0

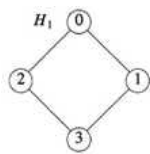


G_3

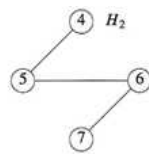


(b) G_3

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

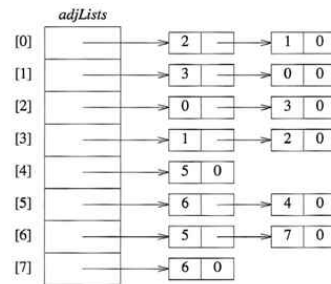


H_1



H_2

G_4

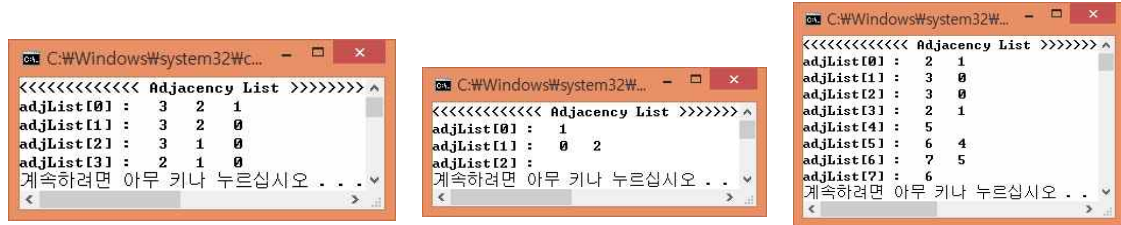


(c) G_4

(2) 실행순서

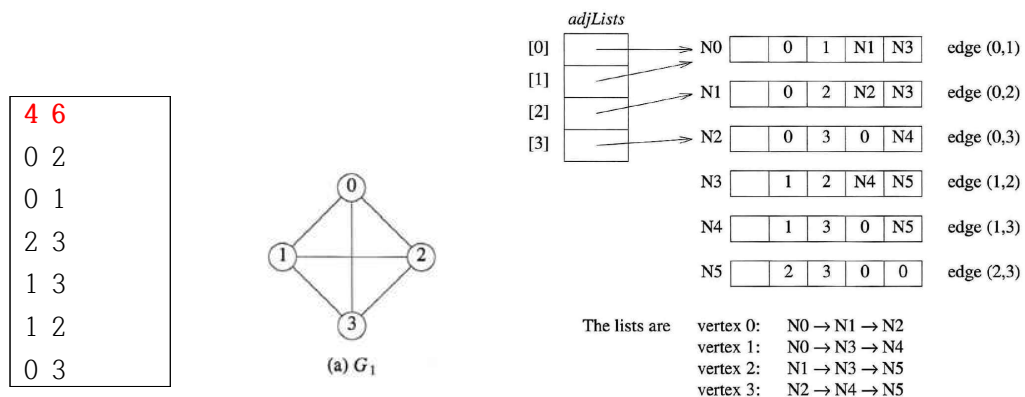
- ① 그래프 종류, 정점, 간선의 수를 입력받음
- ② 그래프 종류에 따라 간선을 하나씩 입력받으면서 인접리스트를 구성함
 - ※ 항상 헤더노드가 가리키는 처음 노드로 입력되게 함
- ③ 각 정점에 대해 부속된 간선(edges incident to a vertex v)을 출력하기

(3) 실행 예 (G1, G3, G4)



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	<p>adjList[i] → ... j → ...</p> <p>adjList[j] → ... i → ...</p>	두 개의 노드
Adjacency Multilists	<p>adjMulList[i] → ... 정점 i만 따라가자! → ...</p> <p>adjMulList[j] → ... 정점 j만 따라가자! → ...</p> <p>... i j ...</p> <p>혹은 (j i)</p>	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 마차가지로 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

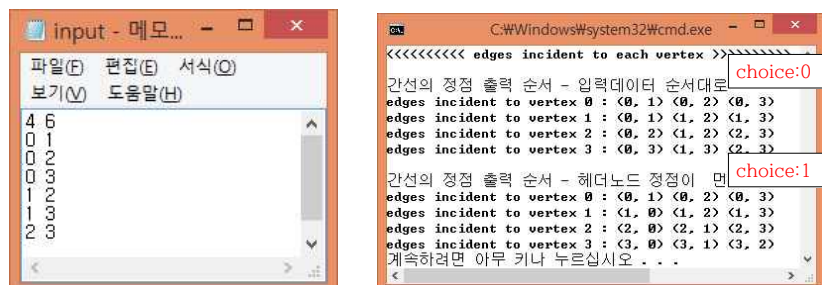
tEdgePointer createEdge(int ver1, int ver2); // create the edge node
void addEdge(tEdgePointer temp, int ver1, int ver2);
void printMultilist(int numVertex, int choice); // choice: 0 or 1
```

(3) 실행순서

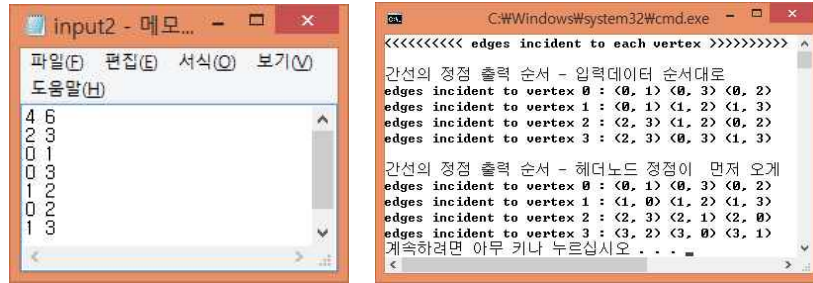
- ① 정점과 간선의 수를 입력받음
- ② 그래프를 구성하는 간선을 하나씩 입력받으면서 adjacency multilist를 구성함
 - ※ 같은 간선이 두 번 입력되지 않음을 가정함
- ③ 각 정점에 대해 부속된 간선(edges incident to a vertex v)을 출력하기
 - ※ 위 1번 문제 adjacency list의 출력 결과와 비교해 보라.

(4) 입력 파일 및 실행 예 (※ 그래프 G1에 대한 세 가지 입력의 경우)

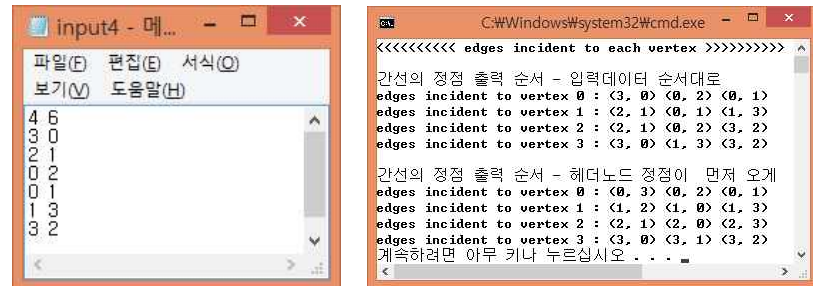
- ① case 1



② case 2



③ case 3



※ 채점기준

- ✓ 간선의 정점 출력 순서 - 입력데이터 순서대로 출력
- ✓ 간선의 정점 출력 순서 - 헤더노드 정점이 먼저 오게 출력

■ 제출 형식

- 제출 : 12월 6일자 채점서버에 소스 업로드(12월 19일까지)
- 실행화면을 캡처하여 한글파일 보고서(DS 4_학번.hwp)에 추가 후 과제에 제출 (12월 19일까지)
- 각 소스파일에 주석처리 추가
 - “학번 이름”
 - “본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

■ 주의

- **소스복사로는 실력향상을 기대할 수 없습니다!!!**
- 먼저 제출한 학생은 남은 시간 동안 자료구조 관련 개인학습을 하거나 동료를 도와 줄 것
- 수강생 끼리 서로 물어보고 논의를 해도 됨
- 채점서버에 제출하지 않는 경우 최종 점수의 0점으로 처리함
(사용법을 잘 모르겠다면 개인적으로 튜터나 TA를 찾아와서 물을 것)