

자료구조응용

17. Adjacency List, Adjacency Multilists (15점)

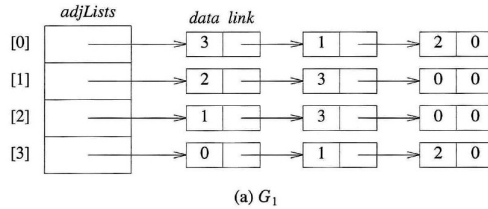
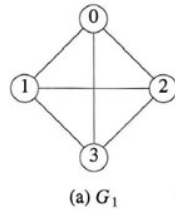
2020.5.13.(수)

1. 다음과 같이 파일 입력을 통해 무방향 그래프(undirected graph)나 방향 그래프(directed graph)를 인접리스트(adjacency-list)로 구성하는 프로그램을 작성하시오.(5점)

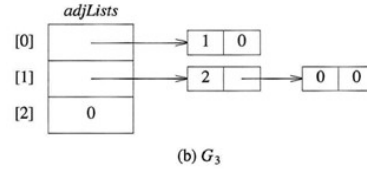
(1) 입력파일(input.txt) 및 자료구조

- ※ 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

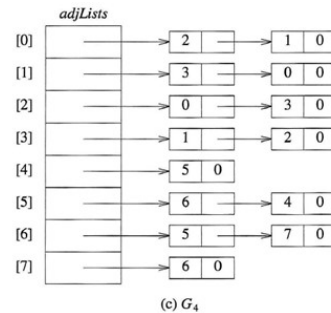
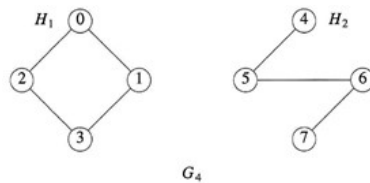
u 4 6
0 1
0 2
0 3
1 2
1 3
2 3



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 부속된 간선(edges incident to a vertex v)을 출력하기

- 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

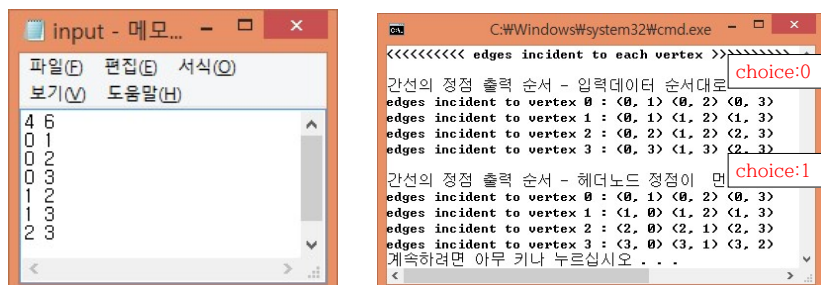
tEdgePointer createEdge(int ver1, int ver2); // create the edge node
void addEdge(tEdgePointer temp, int ver1, int ver2);
void printMultilist(int numVertex, int choice); // choice: 0 or 1
```

(3) 실행순서

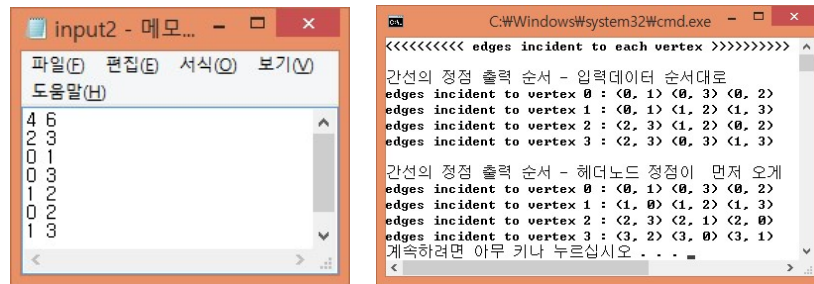
- ① 정점과 간선의 수를 입력받음
- ② 그래프를 구성하는 간선을 하나씩 입력받으면서 multilist를 구성함
※ 같은 간선이 두 번 입력되지 않음을 가정함
- ③ 각 정점에 대해 부속된 간선(edges incident to a vertex *v*)을 출력하기
※ 위 1번 문제 adjacency list의 출력 결과와 비교해 보라.

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

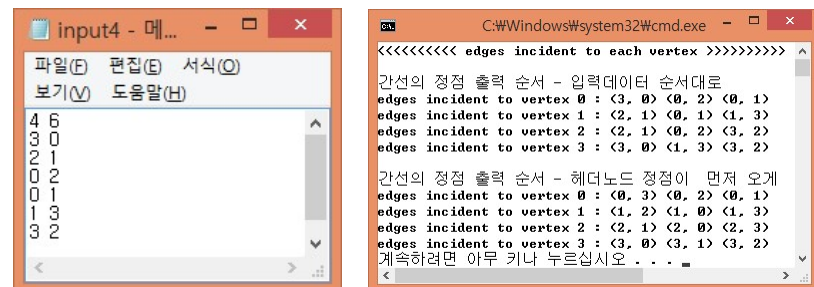
① case 1



② case 2



③ case 3



※ 채점기준

- ✓ 간선의 정점 출력 순서 - 입력데이터 순서대로 출력 (9점)
- ✓ 간선의 정점 출력 순서 - 헤더노드 정점이 먼저 오게 출력 (1점)
- ✓ TA/Tutor는 adjacency list가 아닌 adjacency multilist로 제대로 구현하였는지 소스레벨에서 확인하기 바랍니다.

■ 제출 형식

- 솔루션 이름 : DS 17
- 프로젝트 이름 : 1, 2
- 각 소스파일에 주석처리
“학번 이름”
“본인은 이 소스파일을 복사 없이 직접 작성하였습니다.”
- 솔루션 정리 후 솔루션 폴더를 “학번.zip”으로 압축하여 과제 게시판에 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 1차 마감 : 5월 14일(목) 자정
- 2차 마감 : 5월 15일(금) 자정(만점의 80%)