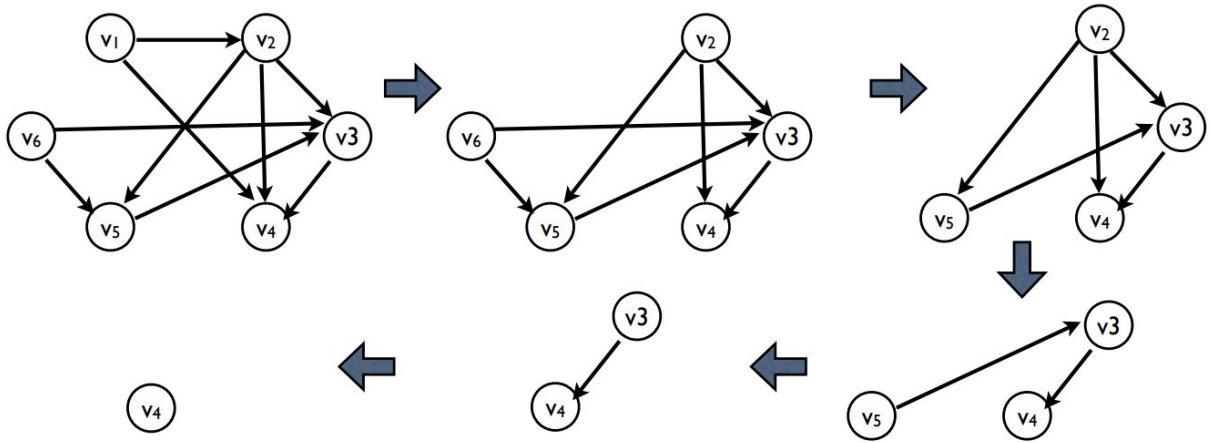
Lab 10: Topological Sorting

Data Structure 2023

Topological Sorting

- Algorithm
  - for each vertex v whose in-degree is zero,
    - print v
    - remove v and its outgoing edges (which leads to decrementing the in-degree value of v's adjacent vertices)
- Use either stack or queue to keep track of vertices with in-degree = 0
- Use adjacency list representation or matrix

Topological Sorting



# of in-dgree

v1	0	0	0	0	0	0
v2	1	0	0	0	0	0
v3	3	3	2	1	0	0
v4	3	2	2	1	1	0
v5	2	2	1	0	0	0
v6	0	0	0	0	0	0
queue	v1, v6	v6, v2	v2	v5	v3	v4
dequeue	v1	v6	v2	v5	v3	v4
queue	v6	v2	none	none	none	none

	v1	v2	v3	v4	v5	v6
v1	0	0	0	0	0	0
v2	0	0	0	0	0	0
v3	0	0	0	0	0	0
v4	0	0	0	0	0	0
v5	0	0	0	0	0	0
v6	0	0	0	0	0	0

Topological Sorting ADT

- **Graph CreateGraph(int* nodes, int n)**
  - Create a graph with n nodes. All input nodes will be positive numbers
- **void InsertEdge(Graph* G, int a, int b)**
  - Insert an edge from node a to node b
- **int* Topsort(Graph* G)**
  - Return the result by topological sorted array
  - Return **NULL** if the graph has a cycle
  - Sort the smaller number key if same priority
- **void DeleteGraph(Graph * G)**
  - Deallocate memory of the Graph

Queue ADT

- **Queue* MakeNewQueue(int X)**
  - Create a new queue with the size of X
- **int Dequeue(Queue* Q)**
  - Insert a new element at the end of the queue
- **void Enqueue(Queue* Q)**
  - Pop the element in the front of the queue
- **int IsEmpty(Queue* Q)**
  - Return 1 if the queue is empty, otherwise, 0
- **void DeleteQueue(Queue* Q)**
  - Deallocate the queue

Topological Sorting ADT

Structure

```
typedef struct _Queue {
    int* key;
    int first;
    int rear;
    int qsize;
    int max_queue_size;
}Queue;
```

```
typedef struct _Graph {
    int size;
    int* node;
    int** matrix;
}Graph;
```

Function

```
Graph* CreateGraph(int* nodes, int n);
void InsertEdge(Graph* G, int a, int b);
void DeleteGraph(Graph* G);
int* Topsort(Graph* G);
Queue* MakeNewQueue(int X);
int IsEmpty(Queue* Q);
int Dequeue(Queue* Q);
void Enqueue(Queue* Q, int X);
void DeleteQueue(Queue* Q);
```

Input & Output Example

```
(base) oknkc8@DESKTOP-NT9MABE:~/CSE2010/lab11$ cat input1.txt
1 2 3 6 5 7
1-2 1-6 2-5 2-6 2-3 3-5 5-6 7-3 7-5
(base) oknkc8@DESKTOP-NT9MABE:~/CSE2010/lab11$ ./lab11_solution input1.txt output1.txt
(base) oknkc8@DESKTOP-NT9MABE:~/CSE2010/lab11$ cat output1.txt
   1  2  3  5  6  7
1  0  1  0  0  1  0
2  0  0  1  1  1  0
3  0  0  0  1  0  0
5  0  0  0  0  1  0
6  0  0  0  0  0  0
7  0  0  1  1  0  0

1 7 2 3 5 6
```

Assignment

- Due
  - **~ 2023.05.24(수) 23:59**
  - Last Commit 기준

- 자세한 내용은 과제 명세 PDF 파일 참고