

# Lab 11 & HW 11

---

*Data Structure*



## *Lab 11 (due on Lab Session)*

1. Do p11\_1.c

## *HW 11 (due on the day before the next Lab Session)*

1. Do p11\_2.c

## *Evaluation criteria*

Category	Evaluation	
p11_1	50	
p11_2	50	
Total	100	

- *Use GCC 4.8 version or GCC 5.4 version.*
- *No score will be given if the gcc version is different.*

# *Lab11 – Topological sorting*

- You should finish p11\_1 (createGraph) during the lab session and submit it to git before you leave.
- For p11\_2 (printTopologicalSorting) you can submit it to the git later.
- Folder name : Lab11
- code name: p11\_1, p11\_2
- No score , if the folder, code names are wrong.
- No score, if it does not use FILE I/O
- Each code will be tested by 5 different input files.
- 10 score for each input, if you don't get the answer you get 0 score.

## *Lab11 – Topological sorting*

**Graph CreateGraph(int[] nodes)** Create a graph with nodes.

**void InsertEdge(Graph G, int a, int b)** Insert a edge.

**void Topsort(Graph G)** Print the graph by topological sort. Sort the smaller number key if same priority.

**Queue MakeNewQueue(int X)** create a new queue with the size of X.

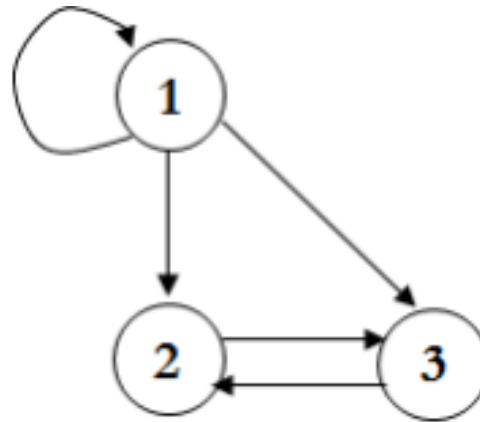
**void Enqueue(Queue\* Q, int X)** a new element at the end of the element in the queue.

**int Dequeue(Queue\* Q)** the node in the front.

# Lab11 – Topological sort

- Structure

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



	1	2	3
1	1	1	1
2	0	0	1
3	0	1	0

# Lab11 – Topological sort

- Structure

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

# Lab 11. Topological sort – *CreateGraph*

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct _Queue* Queue;
typedef struct _Graph* Graph;

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

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

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





## *Lab11. Topological sort – CreateGraph*

- program name : p11\_1.c
- input : Read a set of vertices in the first line and a set of edges in the second line from the given input file. Each line is described below. You may assume that the node is represented by an integer.
- output : the corresponding result in the standard output

# Lab11. Topological sort – Example1

- input file : Lab11\_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
```

- Result

```
1 2 3 6 5 7  
1 0 1 0 1 0 0  
2 0 0 1 1 1 0  
3 0 0 0 0 1 0  
6 0 0 0 0 0 0  
5 0 0 0 1 0 0  
7 0 0 1 0 1 0
```

## *Lab11. Topological sort – Topsort, MakeNewQueue, Enqueue, Dequeue*

- program name : p11\_2.c
- input : Read a set of vertices in the first line and a set of edges in the second line from the given input file. Each line is described below. You may assume that the node is represented by an integer.
- output : the corresponding result in the standard output.

## *Lab11. Topological sort – Example2*

- input file : Lab11\_input2.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
```

- Result

```
1 7 2 3 5 6
```