

Week3

E03-01

Implement algorithms of Breath-First Search(BFS)

- Input :
 - The first line : a number n for n nodes in graph, a number m for m edges in graph, a number i for the beginning node of BFS
 - The following m lines : m edges in an **undirected** graph G
- Output :
 - The layer of each node in BFS tree
- Example :

```
1 Input :
2 5 5 1
3 0 1
4 0 2
5 0 4
6 1 3
7 3 4
8 Output :
9 1 0 2 1 2
```



E03-02

Implement algorithms of connected component with BFS and DFS

- Input :
 - The first line : a number n for n nodes in graph, a number m for m edges in graph, a number i for i th node to find its connected component
 - The following m lines : m edges in an **undirected** graph G
- Output :
 - The first line : connected component found by BFS sorted by iteration order
 - The second line : connected component found by DFS sorted by iteration order
 - Tips : if some nodes can be considered at the same time, please sort them by index, eg. when considering n_1 and n_2 , n_1 should be iterated first.
- Example :

```

1 Input :
2 6 5 1
3 0 1
4 0 2
5 1 3
6 4 5
7 Output :
8 1 0 3 2
9 1 0 2 3

```



E03-03

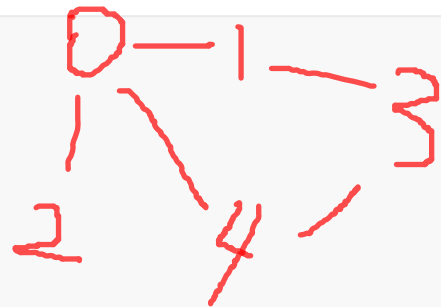
Implement the algorithms of testing bipartiteness

- Input :
 - The first line : a number n for n nodes in graph, a number m for m edges in graph
 - The following m lines : m edges in an **undirected** graph G
- Output :
 - "Yes" if the graph is bipartite graph or "No"
- Example :

```

1 Input :
2 5 5
3 0 1
4 0 2
5 0 4
6 1 3
7 3 4
8 Output :
9 Yes

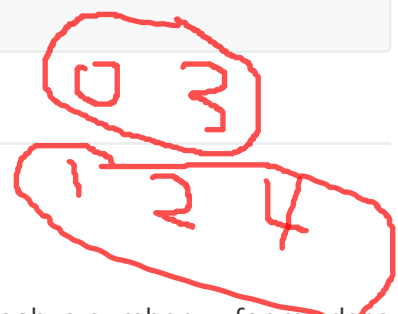
```



E03-04

Implement the algorithms of the topological order

- Input :
 - The first line : a number n for n nodes in graph, a number m for m edges in graph
 - The following m lines : m edges in a **directed** graph G
- Output :
 - The topological ordering of G (sorted by the index of node if the degrees of some nodes are zero at the same time)
- Example :



```
1 Input :  
2 5 5  
3 0 4  
4 1 0  
5 1 3  
6 2 0  
7 3 4  
8 Output :  
9 1 2 3 0 4
```

