Q8.

For this problem,we will implement a **Graph** data structure and a **breadth-first search(BFS)** function.

Graph Specification:

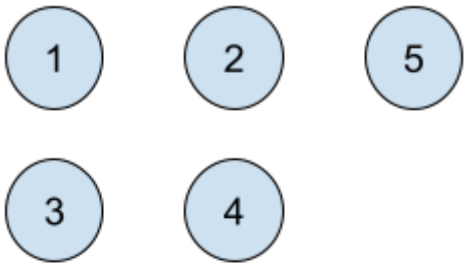
- The graphs are undirected
- Each graph has exactly one connected component
- The starting node for BFS is the first node added to the graph
- The order in which nodes at the same level are visited will follow the order in which the edges were added to the graph

Hint:

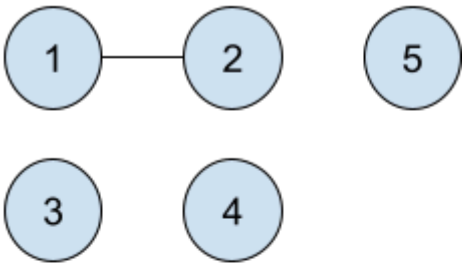
The following is an example of undirected graph

Use **queue** in BFS function

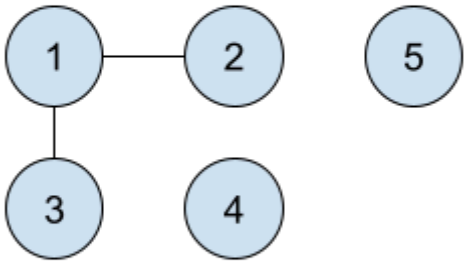
1.Node1~5



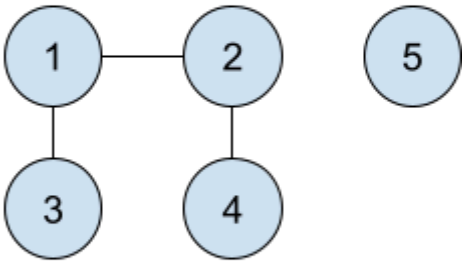
2. add edge(1,2)



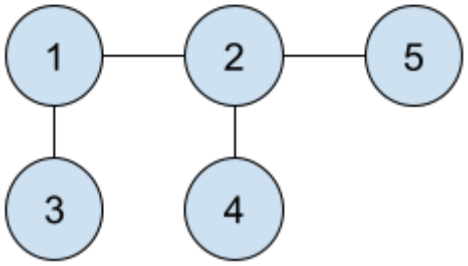
3 add edge(1,3)



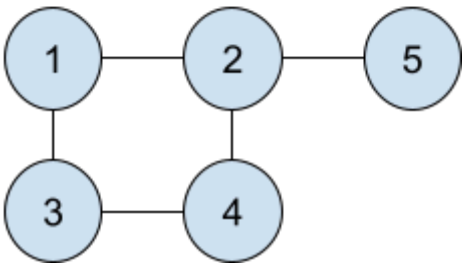
4. add edge(2,4)



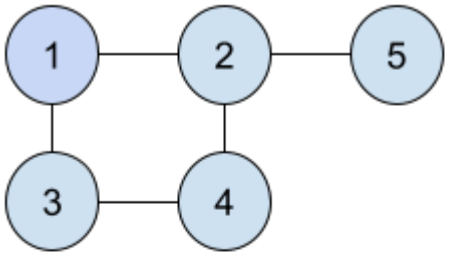
5. add edge(2,5)



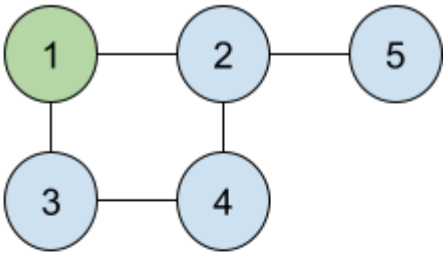
6. add edge(3,4)



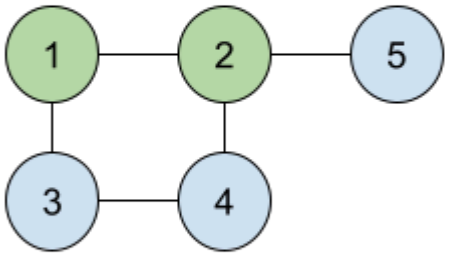
1. q={1}



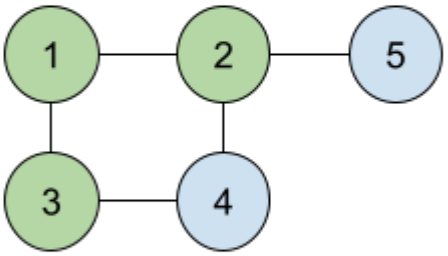
2. q={2,3}



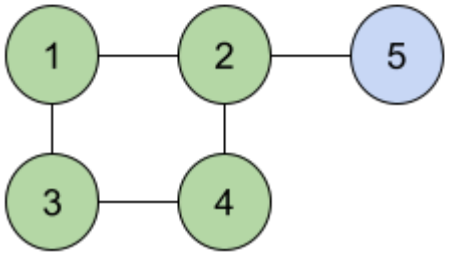
3. q={3}



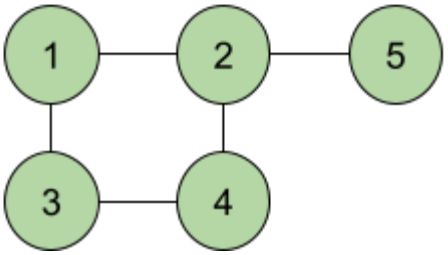
4. q={4,5}



5. q={5}



q={}



Input Format

The first line consists of an integer n which is the number of edges in this case.
Then the following n line are consist 2 integer represent two node's value

Output Format

You need to output the traversal order of nodes visited during a Breadth-First Search (BFS) of the graph

Sample Input 1

5
1 2
1 3
2 4
2 5
3 4

Sample Output 1

1 2 3 4 5

Sample Input 2

```
11
6 3
6 2
4 7
2 7
3 1
1 2
5 1
5 9
4 9
6 8
8 10
```

Sample Output 2

```
6 3 2 8 1 7 10 5 4 9
```