

# k-NN w/ Data Structure

Author: Ricardo S & Billy J



Diberikan sebuah graf *undirected* dan *unweighted* dengan **N** *node* dan **M** *edge*. Tentukan semua *node* yang berada tepat pada jarak **L** dari *node* awal **S** menggunakan *Breadth-First Search* (BFS). Jika tidak ada *node* pada jarak tersebut, keluarkan "Not Found".

## Format Masukan

- Baris pertama: tiga bilangan bulat **N**, **M**, dan **L**
- Baris kedua: satu bilangan bulat **S** — *node* asal pencarian
- **M** baris berikutnya, masing-masing dua bilangan bulat **u** dan **v** menandakan *edge* antara *node* **u** dan **v**

## Format Keluaran

- Cetak semua *node* yang tepat berjarak **L** dari **S**, dengan jarak satu baris atau biasanya menggunakan "\n".
- Jika tidak ada *node* yang memenuhi, cetak: Not Found

## Batasan

- ( $1 \leq \mathbf{N} \leq 10^5$ ,  $0 \leq \mathbf{M} \leq 10^5$ ,  $0 \leq \mathbf{L} \leq \mathbf{N}$ )
- ( $0 \leq \mathbf{S}, \mathbf{u}, \mathbf{v} < \mathbf{N}$ )

## Contoh

Sample Input 0
6 5 2

0  
0 1  
1 2  
2 3  
3 4  
4 5

Sample Output 0

2

Penjelasan Contoh 0



Target level = 2, maka yang dicari adalah *node-node* dengan jarak tepat 2 dari *node start* (0). Dari BFS di atas, hanya node 2 yang berada pada level 2.

Sample Input 1

4 2 2  
0  
1 2  
2 3

Sample Output 1

Not Found

Penjelasan Contoh 1



Target level = 2, maka yang dicari adalah *node-node* dengan jarak tepat 2 dari *node start* (0). Dari BFS di atas, TIDAK ADA *node* yang berada pada level 2.

# k-NN w/ Data Structure

Author: Ricardo S & Billy J



Given an undirected and unweighted graph with **N** nodes and **M** edges. Determine all nodes that are exactly at **L** distances from the initial node **S** using Breadth-First Search (BFS). If there are no nodes at that distance, output "Not Found".

## Input Format

- First line: three integers **N**, **M**, and **L**
- Second line: one integer **S** — the starting node for the search
- The next **M** lines contains two integers **u** and **v** indicating an edge between nodes **u** and **v**

## Output Format

- Print all nodes exactly **L** distances away from **S**, with each distance on a separate line or typically using "\n".
- If no nodes meet the condition, print: Not Found

## Constraints

- ( $1 \leq \mathbf{N} \leq 10^5$ ,  $0 \leq \mathbf{M} \leq 10^5$ ,  $0 \leq \mathbf{L} \leq \mathbf{N}$ )
- ( $0 \leq \mathbf{S}, \mathbf{u}, \mathbf{v} < \mathbf{N}$ )

## Example

Sample Input 0
6 5 2
0
0 1

1 2  
2 3  
3 4  
4 5

Sample Output 0

2

Example 0 Explanation



Target level = 2, so we are looking for nodes that are exactly 2 away from the start node (0). From the BFS above, only node 2 is at level 2

Sample Input 1

6 2 2  
0  
1 2  
2 3

Sample Output 1

Not Found

Example 1 Explanation



Target level = 2, so we are looking for nodes that are exactly 2 away from the start node (0). From the BFS above, there are NO nodes at level 2.