G1.txt

5

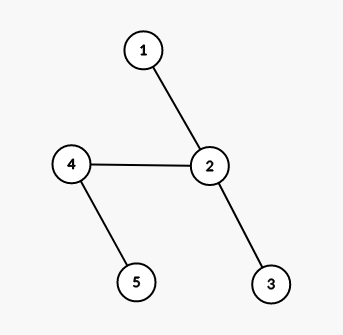
0 1 0 0 0

1 0 1 1 0

0 1 0 0 0

0 1 0 0 1

0 0 0 1 0



Input format:

<operation number(3 or 4)> <file name> <starting vertex>

**DFS**

**Example 1**

Input:

3 G1.txt 1

Output:

3 5

**Example 2**

Input:

3 G1.txt 2

Output:

1 3 5

**Example 3**

Input:

3 G1.txt 4

Output:

1 3 5

**Note:** Order of vertices do not matter here. A node should appear in the output if and only if it becomes a leaf node when the tree is rooted at the starting vertex.

**BFS**

**Example 1**

Input:

4 G1.txt 1

Output:

1 2 3 4 5

**Example 2**

Input:

4 G1.txt 2

Output:

2 1 3 4 5

**Note:** Order of vertices do not matter for vertices at the same distance from the starting vertex.

In example 1, nodes 3 and 4 are both at a distance of 2 from node 1. So, 1 2 4 3 5 is also a correct output. However, any other output apart from these 2 will be incorrect.

Similarly in example 2, the following outputs will be correct as well:

* 2 1 4 3 5
* 2 3 1 4 5
* 2 3 4 1 5
* 2 4 1 3 5
* 2 4 3 1 5

Any other output apart from these 6 will be incorrect.

G2.txt

7

0 1 0 1 0 0 0

1 0 0 0 1 0 1

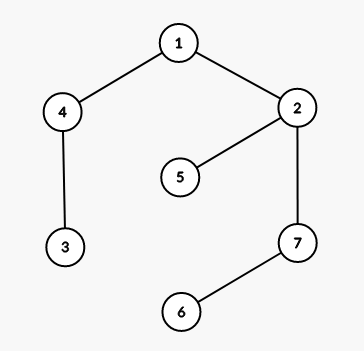
0 0 0 1 0 0 0

1 0 1 0 0 0 0

0 1 0 0 0 0 0

0 0 0 0 0 0 1

0 1 0 0 0 1 0



**Example 3**

Input:

4 G2.txt 1

Output:

1 4 2 3 5 7 8

In this example, while processing the nodes at depth 1 (i.e. nodes 2 and 4) node 3(child of node 4) should be added to the queue by one thread, and nodes 5 and 7(children of node 2) should be added by a separate thread.

At each level, you should use a separate thread for processing each node at that level.