

CSN-261: Data Structures Laboratory

Lab Assignment 10 (L10)

1. For a given adjacency matrix of an undirected graph, you have to count the number of islands. Write a program in Java to do that.

Input

- The adjacency matrix for the graph.
- A group of connected 1's is considered as an island.

Output

Number of islands.

Sample Input

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Sample Output

2

2. Implement a program in Java for B tree containing functions for the following operations.
 - Insertion of an element (no duplicate element is allowed).
 - Deletion of an existing element.
 - Traversing the B Tree using any one of the three traversal algorithms, i.e., in-order, pre-order and post-order.

Input

- First line has an integer z , shows the number of queries.
- Second line has value for the minimum number of child pointers of a B tree node.
- The next z lines have queries and are of the form “i x” (insert x into a B tree) or “d x” (delete x from a B tree).

Output

- The output contains three lines, printing number of split operations, the number of merge operations, and the tree traversal of a B tree produced after the execution of all z queries.
- First line shows the number of split operations performed.
- Second line shows the number of merge operations performed.
- Third line shows the in-order traversal.

Sample Input

```
11
2
i 5
i 9
i 3
i 7
i 1
i 2
i 8
i 6
i 0
i 4
d 9
```

Sample Output

```
4
1
0 1 2 3 4 5 6 7 8
```

3. There are many rooms in a maze that are connected by one-directional paths. If you are in the room, you select (uniformly and randomly) one of the rooms directly connected to it. You will be stuck in a room if you cannot move any further. Assume that there is no back path to the room that you have left. Write a Java program to find the room that you are most likely to get stuck.

Input

- First line: Three integers n (the number of rooms), m (the number of one-directional paths), and r (initial room index).
- Next m lines: Two integers u_i and v_i representing a one-way path from room u_i to v_i .

Output

- Print the index of the room that you are most likely to get stuck. If there are multiple rooms, then print them in the increasing order of indices (space-separated values in a single line).

Constraints

- $1 \leq n \leq 200000$
- $1 \leq m \leq 500000$
- $1 \leq u_i, v_i, r \leq n$

Sample Input

```
5 7 1
1 2
1 3
1 4
1 5
2 4
2 5
3 4
```

Sample Output

```
4
```