CS213: Data Structures and Algorithms **Deadline: 11:30 PM, Feb-20 2021**

Task #5

Instructor: Prof. Sharat Chandran

Important Notes

This task has two programming problems and one written problem.

For this final graded assignment, you are not to use the Internet to get the logic for the stated problems. You can of course use the Internet to figure out simple things like syntax and API, but no Google-fu. All problems are solvable via simple, deliberate impassioned thinking. Maybe not optimally but solvable, and that's the point. As always, if you cannot solve the given problem with the specified constraints do provide solutions that might potentially violate the constraints.

Please (continue to) honour your honor code. Before you start the assignment, write the honor code just like you do in the exam. We want you to feel good about solving these problems, just like it was in Grade XII.

1 Rooks and Queens (Programming)

Should you keep the queen, or should you keep two rooks? This had always puzzled Jonas who is looking at a a non-square chess board with M rows and N columns (chessboards are used to visualize so-called grid-graph models). At a given stage of the game, the cells indicate the location of various pieces; Jonas is interested in the (rook) sweep of these pieces. A (rook) sweep of a board is configuration consisting of the union of all cells reachable from the given (occupied) cells via a rook's move.

The goal of this task is to compute the rook sweep given a binary (Boolean) matrix as input indicating the occupied pieces.

```
Input
3 4
0 0 1 0
0 1 0 1
0 0 0 0

Output
1 1 1 1
1 1 1 1
0 1 1 1
```

Explanation: Observe that the lower bottom cell is marked with a zero since it cannot be reached by any of the given pieces using the rook's move.

1.1 Note

For full credit, the constraint in this task (paradoxically, a a hint!) is that you cannot use any data structure (including variables) to store *anything* other than 0 or 1 for solving this problem (You can surely use index variables in `for` loops and to store M and N).

Expected time complexity: O(M*N).

Expected Auxillary Space: O(1)

Reflection Essay:

1. Explain your efficient solution, and also what false steps you made in case your first solution was not using O(1) space. Otherwise discuss whatever solution you came up with for partial marks.
2. Can you come with a even better algorithm than O(M*N) time complexity and O(1) auxiliary space?

2 Join the dots $\cdots$ in sequence (Written)

You are given a directed acyclic Graph with nodes numbered 1 through n, and you are given a sequence of k distinct vertices $v_1, v_2 \cdots v_k$.

Your task is to find a path containing these vertices in the specified order, if such a path exists. That is this path must start with v_1 and end with v_k and in between the path must first visit v_2 then v_3 then v_4 and so on.

Your goal is to perform this operation in $O(n + m)$ time.

Reflection Essay:

1. You should explain why your algorithm has linear running time and also explain why this algorithm is correct.

Hint: For reduced credit you may describe an $O(k(n + m))$ algorithm. Once you do that, you may able to see how to cut the cost.

3 Alice in Candyland (Programming)

Alice has just discovered Candyland! Candyland has n blocks numbered $1, 2, \ldots, n$ and $n-1$ roads connecting these blocks such that each block is connected to all other blocks. Some blocks have candy shops while others don't. Since Alice is fond of candies, she wants to choose a block to live in such that the total distance of her block from all the blocks having candy shops is minimum possible. Can you help Alice to find such a block?

Input Format

The first line contains two space separated integers n and k (the total number of blocks and the number of blocks having candy shops respectively).

The next $n - 1$ lines contain two space separated integers u and v denoting there is a road connecting blocks u and v.

The last line contains k space separated integers $a_1, a_2, \ldots, a_k$ (the block numbers of blocks having candy shops).

Constraints
$2 \leq n \leq 2 \times 10^5$
$1 \leq k \leq n$
$1 \leq u, v \leq n$

$1 \leq a_i \leq n \;\; \forall i \in [k]$

Output Format

The output should contain a single integer indicating the block number of the block which minimizes the total distance to each block with candy shop. If there are multiple possible answers, print the one having the least block number.

```
Input
5 3
1 2
2 3
2 4
4 5
1 3 5
Output
2
```

Reflection Essay

- Describe your algorithm in detail and explain why it is correct.
- Mention the time complexity of your algorithm and justify it.
- Should Alice necessarily live on a block that has a candy shop?