

Assignment-5

Data Structures and Algorithms Graphs

Due Date: 15 June 2023 11:59 PM

Important Notes

- Doubt Document
- Plagiarism Policy. Please read this before you start the assignment
- Usage of C++ is **NOT** allowed for this assignment.

1 Space Travel

Bob lives in a solar system with n planets and m bridges between two planets, one can only use these bridges to travel to other planets. The planets are numbered from $1, 2, \dots, n$. Bob currently resides on the planet numbered 1, and wishes to travel to all the other planets. He has a notebook, which he uses to keep track of the planets he has visited so far. He initially writes the number 1 on the notebook (since he starts off his journey from planet 1), and whenever he visits a planet which is not recorded on his notebook. He records the number of the planet. He stops after visiting all the planets, and thus his notebook contains a permutation of all the planets.

Bob wants to travel the solar system in such a way that the sequence of nodes being recorded is *smallest* possible sequence. We say that a sequence s_1 is smaller than s_2 if :

- In the first position where the two sequences differ, s_1 has a smaller element than s_2 at that position.

Hence the sequence $\{1, 8, 7, 6, 2, 3, 10, 4, 9, 5\}$ is smaller than the sequence $\{1, 8, 7, 10, 5, 2, 3, 6, 4, 9\}$. Note that we are **NOT** comparing the sequences lexicographically and instead comparing the values at each position.

Help Bob find this smallest possible sequence which can be recorded.

1.1 Input Format

The first line consists of two integers, n and m , indicating the number of planets, and the number of bridges between planets respectively.

The following m lines consists of two values u and v , describing a bridge between the planets (u, v) . Note that these bridges are bidirectional, meaning one can travel both ways using the same bridge.

It is guaranteed that the solar system is connected, meaning that there exists a path from every planet to every other planet.

1.2 Output Format

Output a line containing the smallest sequence $p_1, p_2 \dots p_n$ which Bob can record.

1.3 Constraints

- $1 \leq n \leq 10^5$
- For every bridge $(u, v) : 1 \leq u, v \leq n$
- The solar system is always connected

1.4 Sample Test Cases

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

Input	Output
3 2 1 2 1 3	1 2 3

Explanation for Test Case 1

In the second sample, Bob's optimal travel path could be $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 5$. Therefore, Bob will obtain the sequence $\{1, 4, 3, 2, 5\}$. Notice that we only record the planets in the journal which we have not encountered before.

Explanation for Test Case 2

Here, the optimal travel path would be $1 \rightarrow 2 \rightarrow 1 \rightarrow 3$. Hence the corresponding record on his notebook will be $\{1, 2, 3\}$. Another possible path could be $1 \rightarrow 3 \rightarrow 1 \rightarrow 2$. However the record for it would be $\{1, 3, 2\}$, which is larger than the sequence described previously.

2 Don't use Elevator

Vindya City is rectangular. It can be divided into $N \times M$ grid, each cell (R, C) representing a 1-acre square plot. ($1 \leq R \leq N$, $1 \leq C \leq M$).

Each cell (R, C) has height $H(R, C)$ given to it .

From cell (R, C) , a person can move only to cells $(R+1, C)$, $(R-1, C)$, $(R, C+1)$ and $(R, C-1)$. (given those positions lie in the Grid).

For moving between two cells of **different** heights, usage of the elevator is needed. The elevator is not needed to move between two cells with the same height.

Starting at $(1, 1)$, what is the minimum number of times the elevator needs to be used to reach (N, M)

2.1 Input Format

- First line contains the number of test cases T . $1 \leq T \leq 5$
- First line of each test case contains N, M defining dimension of Vindya city $1 \leq N, M \leq 10^3$
- Next N lines of the test case contain M integers representing the height of cells in the grid.
 - The j^{th} element at i^{th} line represents height of cell (i, j) .

2.2 Output Format

- For each test case, output one integer representing the minimum number of times the elevator needs to be used to reach (N, M) from $(1, 1)$.

2.3 Sample test-case

Input	Output
<pre> 4 2 2 1 1 1 1 2 3 1 2 3 4 5 6 6 6 1 2 1 3 3 3 1 1 1 3 5 3 1 7 4 5 3 3 1 9 2 6 4 4 11 10 12 13 4 8 11 10 12 13 4 4 5 5 1 2 2 2 3 1 2 1 3 3 1 1 1 3 3 1 5 6 3 8 3 4 7 4 4 </pre>	<pre> 0 3 2 2 </pre>

2.4 Explanation:

2.4.1 test-case 1

- All cells' height is 1, so all cells in any path from (1,1) to (2,2) are of the same height. So use of an elevator is never needed.

2.4.2 test-case 2

- All cells' height is different. so moving between any two cells requires the usage of an elevator.
- The minimum number of cells visited while moving from (1, 1) to (2, 3) is 4 (including (1,1)).
- So usage of the elevator is required all three times while moving from one cell to another.
- one such path is $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3)$

2.4.3 test-case 3

- The path with the minimum times of using the elevator is $(1, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (2, 3) \rightarrow (1, 3) \rightarrow (1, 4) \rightarrow (1, 5) \rightarrow (1, 6) \rightarrow (2, 6) \rightarrow (3, 6) \rightarrow (4, 6) \rightarrow (4, 5) \rightarrow (5, 5) \rightarrow (6, 5) \rightarrow (6, 6)$.
- The path is shown by a green color in the image below, the cells colored with dark green $\{(1,4),(4,6)\}$ are the cells for which usage of the elevator was needed to enter the cell.
- So, usage of the Elevator was required twice.

- There are some more paths that require the usage of the elevator only twice. You can also check that there are no paths from (1,1) to (6,6) which require an elevator only twice.

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

Indexing of Vindya

1	2	1	3	3	3
1	1	1	3	5	3
1	7	4	5	3	3
1	9	2	6	4	4
11	10	12	13	4	8
11	10	12	13	4	4

Passwords with Path (In green)

2.4.4 test-case 4

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

2.5 Scoring and Tip

- Batch 1: $T = 1$, $1 \leq N, M \leq 10$ (20 Points)
- Batch 2: $T = 1$, $1 \leq N, M \leq 100$ (30 Points)
- Batch 3: $T = 1$, $1 \leq N, M \leq 1000$ (40 Points)
- Batch 4: $T \leq 5$, $1 \leq N, M \leq 1000$ (10 Points)
- use short int for Height. Taking input will be faster.