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 $1, 2, \ldots, n$. 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 visisted 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 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 $p_1, p_2 \ldots p_n$ which Bob can record.

1.3 Constraints

- $1 \leq n, m \leq 10^5$
- 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
4	
2 2 1 1 1 1	0
2 3 1 2 3 4 5 6	3
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	2
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	2

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.

3 Cultivating Vinland

Thorfinn is a trader and explorer looking for the mysterious land called Vinland. When Thorfinn finally finds Vinland, he decides to settle there and cultivate the land. Vinland is a rectangular piece of land that can be divided into a grid of N by M square patches, each of unit side length. Not all patches of land are fertile. Crops can only be grown on fertile patches of land. As such, patches of land need to be selected carefully for cultivation.

Each selected patch needs to be protected from animals by building a fence. An animal located at a patch (i, j) can move up, down, right or left i.e. to $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, or $(i-1, j+1)$. A patch is said to be enclosed if no animals can enter the patch. If a patch of land is not completely enclosed, the crops will be destroyed by grazing animals, and the land will not yield any profit. Building a unit length of fence costs 1 silver coin. A unit area of enclosed land will yield crops that can be sold for 4 silver coins. You may assume that **no animals are initially present on fertile land**.

As the only mathematician in Thorfinn's group, it falls upon you to help Thorfinn select land to cultivate. You need to **select one continuous piece of fertile land** that yeilds the maximum profit (profit = crop earings - cost of fencing)

3.1 Input Format

The first line consists of three space-separated integers, N, M and R. N, M denoting the size of the arena, and R denoting the number of fertile patches in the land.

Then, R lines follow. Each containing 2 integers i, j indicating positions of the fertile patches.

3.2 Output Format

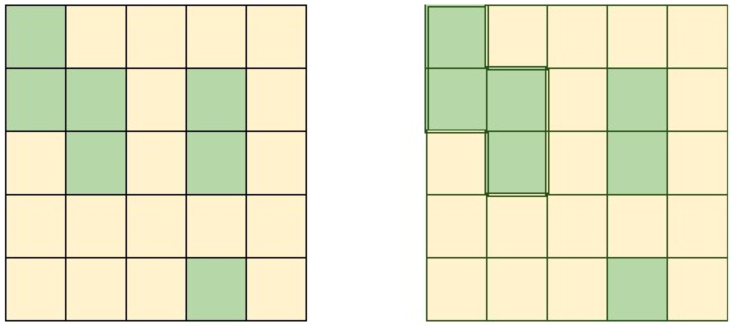
Output should contain a single integer representing the maximum profit that can be made.

3.3 Sample Test Cases

Input	Output
5 5 7 1 1 2 1 2 2 2 4 3 2 3 4 5 4	6

Input	Output
5 4 8 1 1 1 2 1 3 2 2 4 3 4 4 5 3 5 4	8

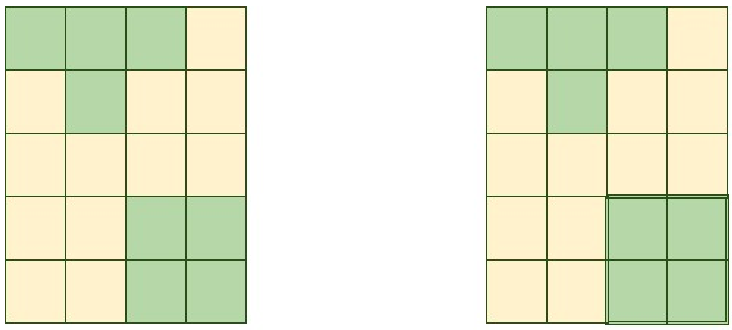
Explanation of sample testcases:
Testcase-1 :



Map Selected Patches

The selected region uses fence of length 10 units includes an area of 4 sq. unit. Yielding a profit of $4 \times 4 - 10 = 6$.

Testcase-2 :



Map Selected Patches

The selected region uses fence of length 8 units includes an area of 4 sq. unit. Yielding a profit of $4 \times 4 - 8 = 8$.

3.4 Constraints

- All input values are integers
- $1 \leq N \times M \leq 10^6$
- $1 \leq N, M \leq 10^6$
- $0 \leq R \leq N \times M$

4 Fortify

You're in an arena consisting of $N \times M$ cells shaped in the form of a $N \times M$ grid. The cell located on the i^{th} row from the top and the j^{th} column from the left is represented by the pair (i, j). You're currently standing in the cell (N, M) and spot a monster currently on cell $(1, 1)$. Thankfully, you've got *fortification spells* to save yourself.

The monster can move into any of the cells adjacent to the one it's currently on, any number of times. To stop it, you may select some cells to *fortify* them. Note that the selected cells can be any valid subset of the grid, and so, need not be adjacent to one another. The monster cannot enter a fortified cell, nor can it leave one if it happens to be already on a fortified cell.

However, some cells are harder to fortify than others. Formally, it costs $A_{i,j}$ energy to fortify the cell (i, j). So, the total energy you'd need to spend is a sum over the costs for all the cells you pick. Note that once a cell is fortified, it stays fortified forever.

Find the minimum total energy you'd have to spend in order to ensure that the monster does not get to you.

Note that two cells (a, b) and (x, y) are adjacent iff $|a - x| + |b - y| = 1$. Note that fortifying the cell (N, M) is a valid solution since it stops the monster from entering the cell. Similarly, fortifying the cell $(1, 1)$ is also always a valid solution since the monster can now never leave its starting cell. But do note that either of these may not be the optimal approach to stopping the monster with respect to the cost incurred.

4.1 Constraints

- All input values are integers
- $2 \leq N \times M \leq 2 \times 10^5$
- $1 \leq N, M \leq 2 \times 10^5$
- $1 \leq A_{i,j} \leq 10^3$ for all $1 \leq i \leq N, 1 \leq j \leq M$
- $1 \leq T \leq 100$
- Sum over $N \times M$ across all test-cases does not exceed 10^6

4.2 Input Format

The first line consists of a single integer T that denotes the number of test-cases. Then, the description of each test-case follows.
The first line of each test-case consists of two space-separated integers, N and M, denoting the size of the arena. Then, N lines follow describing the costs for each row.
The i^{th} line $(1 \leq i \leq N)$ consists of M space-separated integers $A_{i,1}, A_{i,2}, \ldots, A_{i,M}$ denoting the costs of cells on the i^{th} row of the arena.

4.3 Output Format

For each test-case, output on a single line, the minimum total energy you'd have to spend in order to ensure that the monster does not get to you. It can be shown that it is always possible to do so.

4.4 Sample Test Cases

Input	Output
1 2 2 7 4 4 7	7

Input	Output
1 3 3 10 1 5 11 3 1 2 7 6	6

4.5 Explanation of samples test-cases

Test-case 1: In the first case, it is optimal to fortify the cell $(1,1)$ incurring a cost of 7. It can be shown that any other approach of fortifying incurs a larger cost.
Note that fortifying the cell $(1,1)$ which is the starting cell is sufficient to stop the monster since the monster *cannot enter a fortified cell, nor can it leave one if it happens to be already one a fortified cell.*

Test-case 2: It can be shown that the optimal plan of fortification is to fortify the cells $(2,2),(2,3),(3,1)$ incurring a total cost of 6. Alternatively, one could also incur the same cost by fortifying the cell $(3,3)$. Either of these solutions are optimal.