CS29003 ALGORITHMS LABORATORY
Assignment 8 : Graph Algorithms
Date: 29 – Oct – 2020

Consider a $N \times N$ grid of squares (or blocks) with co-ordinates ranging from $(0,0)$ to $(N-1, N-1)$. Each value of the grid (grid[i][j]) represents the elevation at that particular block (i, j). When you start to move at time $t = 0$, the blocks rise such that at time t the elevation of $(i, j)^{th}$ block would be $max(grid[i][j], t)$. You can only move to your adjacent blocks (*top*, *bottom*, *left* or *right*) if and only if the elevation of both the blocks individually are at most t. You can move infinite distance in zero time, but you must stay within the boundaries of the grid during your move.

Initially you are standing at $(Sx, Sy)^{th}$ block.

- ▷ What is the least time in which you can reach the $(Fx, Fy)^{th}$ block?
- ▷ Print the path travelled to reach $(Fx, Fy)^{th}$ block.
- ▷ Also print the number of blocks traversed to reach $(Fx, Fy)^{th}$ block.

There are many ways to solve this problem. **Solve using BFS or DFS**. Consider the blocks as nodes and edges run to the four neighboring blocks (*top*, *bottom*, *right* and *left*). Usage of STL is **not allowed**.

Input

- ▷ First line of input contains the integer N denoting the size of the grid.
- ▷ Next N lines contain N integers each where each integer X_{ij} represents the initial elevation of $(i, j)^{th}$ block in the grid.
- ▷ The next line contains two pairs of integers (Sx, Sy) and (Fx, Fy), representing your start and target positions, respectively.

$$0 \leqslant Sx, Sy \leqslant N-1 \qquad 0 \leqslant Fx, Fy \leqslant N-1$$

Sample Input and Output

Test Case 1

Input:

```
 5
 0   1   2   3   4
24  23  22  21   5
12  13  14  15  16
11  17  18  19  20
10   9   8   7   6
 0   0   4   4
```

Output:

Minimum time taken is: 16

The Path to reach from (0,0) to (4,4) is:
(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (1, 4), (2, 4), (2, 3), (2, 2), (2, 1), (2, 0), (3, 0), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4)
The Number of Blocks traversed are: 17

<u>Explanation:</u>
At $t = 0$, you start at $(0, 0)^{th}$ block. The neighbouring block with smallest height is (0, 1) which is 1. So, at $t = 1$, the height of your standing block ($(0, 0)^{th}$ block) will become 1. Therefore you can move from $(0, 0)^{th}$ block to $(0, 1)^{th}$ block at $t = 1$.

At $t = 5$, you will be at $(1, 4)^{th}$ block whose elevation is 5. The elevations of the blocks at $t = 5$ is shown below. The bold numbers indicate the path travelled till now.

5	**5**	**5**	**5**	**5**
24	23	22	21	**5**
12	13	14	15	16
11	17	18	19	20
10	9	8	7	6

At $(1, 4)^{th}$ block, the neighbour with smallest height is $(2, 4)^{th}$ block which is 16. So you have to wait until $t = 16$ to move to $(2, 4)^{th}$ block. The elevations of the blocks at $t = 16$ are:

16	16	16	16	16
24	23	22	21	16
16	16	16	16	16
16	17	18	19	20
16	16	16	16	16

Now at $t = 16$, the elevation at $(1, 4)^{th}$ block would be 16. Now you can move to any block of height 16. Therefore you can reach the $(4, 4)^{th}$ block at $t = 16$.

16	**16**	**16**	**16**	**16**
24	23	22	21	**16**
16	**16**	**16**	**16**	**16**
16	17	18	19	20
16	**16**	**16**	**16**	**16**

The final route is marked in bold above. You need to wait until time 16 so that $(0, 0)$ and $(4, 4)$ are connected.

Test Case 2

Input:

10

55	33	29	78	47	62	60	79	41	54
34	16	93	64	38	46	91	8	40	65
22	74	12	70	28	80	90	32	6	45
23	49	85	52	11	56	83	5	36	95
31	48	14	89	76	82	19	26	97	63
0	75	9	77	2	51	94	7	71	99
35	81	44	87	43	18	67	17	13	57
92	53	37	39	20	88	15	68	24	66
27	69	84	3	72	10	61	30	50	58
73	96	98	25	4	21	86	1	59	42

0 0 9 9

Output:

Minimum time taken is: 61

The Path to reach from (0,0) to (9,9) is:

(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (4, 1), (4, 2), (5, 2), (6, 2), (7, 2), (7, 3), (8, 3), (9, 3), (9, 4), (9, 5), (8, 5), (8, 6), (8, 7), (9, 7), (9, 8), (9, 9)

The Number of Blocks traversed are: 21

Explanation:

55	33	29	78	47	62	60	79	41	54
34	16	93	64	38	46	91	8	40	65
22	74	12	70	28	80	90	32	6	45
23	49	85	52	11	56	83	5	36	95
31	**48**	**14**	89	76	82	19	26	97	63
0	75	**9**	77	2	51	94	7	71	99
35	81	**44**	87	43	18	67	17	13	57
92	53	**37**	**39**	20	88	15	68	24	66
27	69	84	**3**	72	**10**	**61**	**30**	50	58
73	96	98	**25**	**4**	**21**	86	**1**	**59**	**42**

The final route is marked in bold. You need to wait until time 61 so that $(0, 0)$ and $(9, 9)$ are connected.

BONUS QUESTION: All students can attempt. Will be checked only when you are at borderline of grades at the end of the semester.

Use **Dijkstra's Algorithm** to find the least time to reach $(\mathtt{Fx}, \mathtt{Fy})^{th}$ block. Write the code in function `leastTimeDijkstra`. Usage of STL is **not allowed**.

Submission

Please note that your submissions will not be evaluated unless you follow the below specified file naming convention for the program file. <ROLLNO(IN CAPS)>_G<Group_No>_Assign<Assign_No>.c/cpp

Eg: 18CS30004_G03_Assign8.c / 18CS30004_G03_Assign8.cpp