

COSC2430 hw3: Finding the Smallest Sequence

1. Introduction

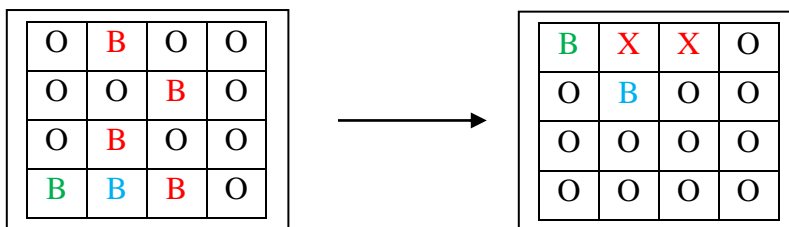
Given a matrix as input, you will need to implement a C++ program to play a mini game and find the smallest sequence that will solve the matrix. This homework will focus on stack and queue implementation. When submit your assignment, please name the folder on the server as “hw3”.

2. Input files

- The first line in the input will contain two integer row and col, indicate the size of the given matrix.
- The rest of the input will contain the matrix that need to be solved.
- Each element will be separated by a space.
- There will be no redundant spaces or empty lines.
- There will be no empty input.

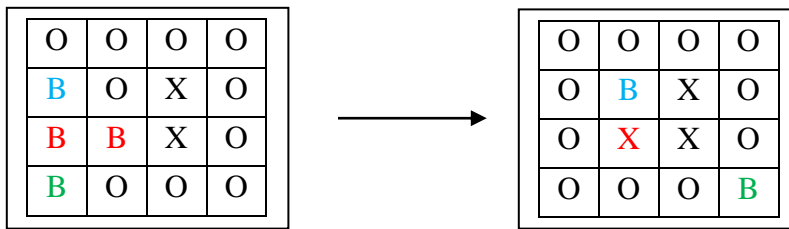
3. Operations

- There are four operation that need to be considered when solving the matrix:
 - o Shift up: shift all the elements in the matrix upward (corresponding to 1)
 - o Shift right: shift all the elements in the matrix to the right (corresponding to 2)
 - o Shift down: shift all the elements in the matrix downward (corresponding to 3)
 - o Shift left: shift all the elements in the matrix to the left (corresponding to 4)
- Elements in the matrix will either be ‘O’, ‘X’, or ‘B’.
 - o ‘O’ will represent an open path in the matrix.
 - o ‘X’ will act as a barrier when shifting the matrix.
 - o ‘B’ will represent a bomb that need to be detonate.
- In order to solve the matrix, all the bomb (B) in the matrix will need to be detonated.
- To detonate a bomb, they need to collide into another bomb by shifting the matrix up, right, down, or left. Which two bomb collided first will be detonated.
- The matrix will always have even number of bomb (B).
- When two bomb detonated, a barrier (X) will be created at the index of detonation.
 - o Ex: Shift up



When shifting up, the bomb (B) marked in red collided and detonated, created two barriers (X).

- When shifting the matrix, all the barrier (X) will be stationary and will not shift.
- A bomb (B) will stop shifting when it meets a barrier (X).
 - o Ex: Shift right



4. Output files

- Output the sequence of moves that detonate all the bomb and solve the matrix.
- If there are multiple sequence that solve the matrix (ex: 124, 211, 3214), output the smallest one.
- If the matrix is already solved without any moves, output “0” for the sequence.

5. Example

- Input1: the moves sequence to solve the matrix is 12 (shift up then shift right)

Given

```

0 0 0 B 0
B 0 0 0 0
0 B 0 0 0
0 0 0 0 B

```

Shift up

```

B B 0 B B
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

```

Shift right

```

0 0 0 X X
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

```

- Input2: the moves sequence to solve the matrix is 141 (up, left, up)

Given

0	0	0	B	0
X	B	0	0	0
0	X	0	0	0
B	0	X	0	B
0	B	0	0	0
0	0	0	0	B

Shift up

0	B	0	B	X
X	0	0	0	0
B	X	0	0	0
0	B	X	0	0
0	0	0	0	0
0	0	0	0	0

Shift left

X	0	0	0	X
X	0	0	0	0
B	X	0	0	0
B	0	X	0	0
0	0	0	0	0
0	0	0	0	0

Shift up

X	0	0	0	X
X	0	0	0	0
X	X	0	0	0
0	0	X	0	0
0	0	0	0	0
0	0	0	0	0

- Input3: the moves sequence to solve the matrix is 1341 (up, down, left, up)

Given

```

0 0 0 B 0 X X 0 0 0
X B 0 0 0 0 0 0 X 0
0 0 0 0 X 0 0 0 B 0
B 0 X 0 B 0 0 X 0 B
0 0 B 0 X 0 0 0 0 0
0 0 X 0 0 0 B 0 B 0
0 0 0 0 B 0 X 0 0 X
B 0 X 0 0 0 0 B 0 0
0 0 0 0 B 0 0 0 X 0
0 X 0 0 B 0 0 0 0 X

```

Shift up

```

0 B 0 B 0 X X 0 0 B
X 0 0 0 0 0 B 0 X 0
X 0 0 0 X 0 0 0 X 0
0 0 X 0 B 0 0 X 0 0
0 0 B 0 X 0 0 B 0 0
0 0 X 0 X 0 0 0 0 0
0 0 0 0 B 0 X 0 0 X
0 0 X 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 X 0
0 X 0 0 0 0 0 0 0 X

```

Shift down

```

0 0 0 0 0 X X 0 0 0
X 0 0 0 0 0 0 0 X 0
X 0 0 0 X 0 0 0 X 0
0 0 X 0 B 0 0 X 0 0
0 0 B 0 X 0 0 0 0 0
0 0 X 0 X 0 B 0 0 B
0 0 0 0 0 0 X 0 0 X
0 0 X 0 0 0 0 0 0 0
0 B 0 0 0 0 0 0 X 0
0 X 0 B B 0 0 B 0 X

```

Shift left

```
0 0 0 0 0 X X 0 0 0
X 0 0 0 0 0 0 0 X 0
X 0 0 0 X 0 0 0 X 0
0 0 X B 0 0 0 X 0 0
B 0 0 0 X 0 0 0 0 0
0 0 X 0 X X 0 0 0 0
0 0 0 0 0 0 X 0 0 X
0 0 X 0 0 0 0 0 0 0
B 0 0 0 0 0 0 0 X 0
0 X X B 0 0 0 0 0 X
```

Shift up

```
0 0 0 X 0 X X 0 0 0
X 0 0 0 0 0 0 0 X 0
X 0 0 0 X 0 0 0 X 0
X 0 X 0 0 0 0 X 0 0
0 0 0 0 X 0 0 0 0 0
0 0 X 0 X X 0 0 0 0
0 0 0 0 0 0 X 0 0 X
0 0 X 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 X 0
0 X X 0 0 0 0 0 0 X
```

6. Requirements

Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. Code that is copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc. will be treated as copy) will be detected and result in "0" in this homework. The limit is 50% similarity.

7. Turn in your homework

Homework 3 needs to be turned in to our Linux server, follow the link here

https://rizk.netlify.app/courses/cosc2430/2_resources/

Make sure to create a folder under your root directory, name it "hw3" (case sensitive), copy all your .cpp and .h file to this folder, "ArgumentManager.h" need to be included as well.

PS: This document may have typos, if you think something illogical, please email TAs for confirmation.