# Assignment 2-Saddle Point

Problem Statement:

An m x n matrix is said to have a saddle point if some entry a[i][j] is the smallest value in row i and the largest value in j. Write C/ C++ function that determines the location of a saddle point if one exists.

## Cocepts Used

* Saddle Point
* Implementation in a Matrix

## Theory of Concepts

### What is a Saddle Point?

In mathematics, a saddle point is a point in the domain of a function where the slopes (derivatives) of orthogonal function components defining the surface become zero (a stationary point) but are not a local extremum on both axes. The saddle point will always occur at a relative minimum along one axial direction (between peaks) and where the crossing axis is a relative maximum.

The name derives from the fact that the prototypical example in two dimensions is a surface that curves up in one direction, and curves down in a different direction, resembling a riding saddle or a mountain pass between two peaks forming a landform saddle. In terms of contour lines, a saddle point in two dimensions gives rise to a contour graph or trace that appears to intersect itselfsuch conceptually might form a ’figure eight’ around both peaks; assuming the contour graph is at the very ’specific altitude’ of the saddle point in three dimensions.

### Implementation in a Matrix

A saddle point is an element of the matrix such that it is the minimum element for the row to which it belongs and the maximum element for the column to which it belongs. Saddle point for a given matrix is always unique.

You can determine it by going through the matrix in row-order and: creating an array to store the current-column maximum storing a current-row-minimum on the fly and store it in an array too when you are done with this you can compare if an index occurs in both at the same time so that you have an index which is both the column-max and row-min.

## Algorithm

Step 1: Start

Step 2: Take matrix from user and store it in mat.

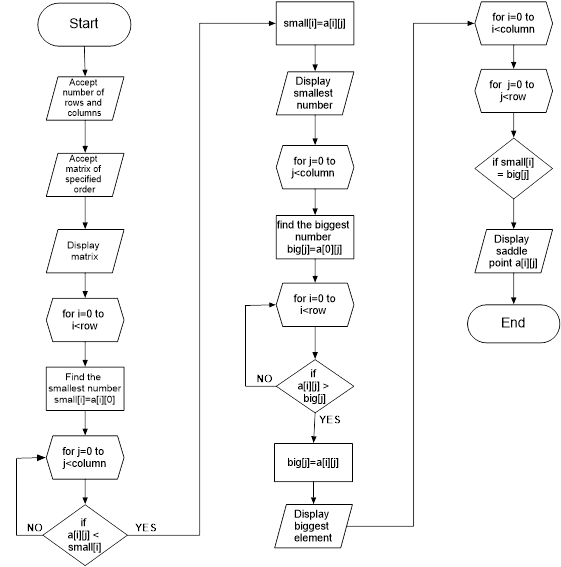
Step 3: Find the minimum element from each row and store in min[] array.

Step 4: Find maximum element from each column and store in max[] array. Step 5: Find maximum element from the min[] array and store in maximum. Step 6: Find minimum element from the max[] array and store in minimum.

Step 7: Compare minimum and maximum

Step 8: If they are same print saddle point exist Step 9: Else print saddle point does not exist Step 10: Stop.

## Flowchart



Output:

