Solution 1

function s = saddle(M)

% Create logical vector that are true for each saddle condition separately

minLocs = M <= min(M, [], 1);

maxLocs = M >= max(M, [], 2);

% Find the indices where both conditions are true!

[row, col] = find(minLocs & maxLocs);

% If the input is a row vector, row and col returned from the find

% function need to be transposed to fit the output format

if isrow(M)

s = [row', col'];

else

s = [row, col];

end

end

Solution 2

function s = saddle(M)

[r, c] = size(M);

% Initialize the saddle points to an empty array

s = [];

% Check the dimensions to see if input is a row or column vector

if r > 1

cols = min(M);

% find the min value in each column if more than 1 row

else

cols = M;

% vector is a special case, min would give a single value

end

if c > 1

rows = max(M');

% find the max value in each row

else

rows = M;

% vector is a special case, max would

%give a single value

end

for ii = 1:c

% visit each column

for jj = 1:r

% and each row, that is, each element of M

if M(jj,ii) == cols(ii) && M(jj,ii) == rows(jj)

% if both conditions hold

s = [s; jj ii];

% saddle point! Let's add it!

end

end

end