ŦPe Short and tPe TalT

R x C students sit in an array of R rows by C columns. TPe shortest 92 eac
row stands up and tPe tallest of tPese A remain1 standing. TPe tallest student

Problems

1. Find the positions of all Saddle Points.
2. Find just one Saddle Point and its position

Let

tf.

MinRow(i) = Minimum of Row I

MaxCol(j) = Maximum of Col J

Saddle_Pt(A(i,j)) ≡ A(i,j) = MinRow(i) ∧ A(i,j) = MaxCol(j)

Q.e. Saddle_Pt(A(i,j)) ≡ MinRow(i) = MaxCol(j)

To find all Saddle-Points in A, find all (i,j) s.t. MinRow(i) = MaxCol(j)

To find just one, we could start by finding all and exit having found the first.

But, we consider an alternative solution which will justify the claim in the 'The Short and the Tall" that if A=B then we have a Saddle-Point.

Notation:

Let f be a function. (Max k | M ≤ k ≤ n : f(k)) = (Min k | M ≤ k ≤ n : f(k)) ≡ (Exists k | 1 ≤ k ≤ ν & M=φ(κ)) ∧ ≤ k ≤ n : f(k)≤M)

Similarly for (Min k | 1 ≤ k ≤ n : f(k))

Theorem 1.

if A(i,j) and A(s,t) are Saddle-Points then A(i,j) =

Proof:

A(i,j) = Min of row I

A(i,R) = MinRow(i) as Saddle_Pt(A(i,R))
also A(i,R) = MaxCol(R) as Saddle_Pt(A(i,R))

Show for all i≤k≤ R, MinRow(k) ≤ A(i,R)

MinRow(k) ≤ A(k,R) -- Min of row k,
≤ A(i,R) -- Max of col j

tf. A(i,R) = (Max k | 1≤k≤R : MinRow(k))

SQmilarly,
A(i,R) = (Min k | i≤k≤R : MaxCol(k))

roof

Theorem 3.

Let MinRow(mx) = (Max k | 1 0.3.76 0 TD /F3 12 Tf -0.108 Tc 0 Tw (£) Tj 6.48 0 TD /F2 1
Show Saddle_Pt(A(mx,mn)).

MinRow(mx) = A(mx,R) some j: 1≤ j≤C
also MaxCol(mn) = A(i,mn) some i: 1≤i≤R

Consider A(mx,mn)
A(mx,R) A(mx,mn)

Theorem 2.

If Saddle_Pt(A(i,R)) then A(i,R) = (Max
also
If Saddle_Pt(A(i,R)) then A(i,R) = (Min

In Eiffel,

class

```
all_saddle (m: MATRIX [INTEGER]) is

    local
        mVr, Uxc: VECTOR [INTEGER];
        tr: MATRIX [INTEGER];
        Q, j: INTEGER
    do
        UVr := U.min_row;
        tr := m.transpose;
        mxc := tr.max_row;
        from

        untQl
            i > U.rows

            from
                j := 1

                j > m.cols
            loop

                    print_saddle (m, Q, j)

                j := j + 1
            end ;
            i := i + 1
        end
    end ;

print_saddle (m: MATRIX [INTEGER]; Q, j: INTEGER)is
    do
        Qput_integer (m.item (Q, j));
        io.p                u                t                _                $
        io.put_integer (Q);

        io.put_integer (j);
        Qput_Vew_Tine
    end ;
```

```
i := 1
```



```
loop
        -- Max of Cols of M
Q t UVr.item (Q) = mxc.item (j) then
and character (' ');
```

```
file2matrix (fname: STRING) Qs
        -- Input from file, fname, into the matrix, mat.
        -- First 2 numbers give #rows and #cols
    Tocal
        in_file: PLAIN_TEXT_FILE;
        i, j, r, c: INTEGER;
        x: INTEGER
    dW
        !! in_file.make_open_read (fname);
        Qn_file.read_integer;
        r := in_file.Tast_integer;
        in_file.read_integer;
        c := in_file.Tast_integer;
        !! mat.make (r, c);

            i := 1
        until
            i > r
        Toop
            from
                j := 1
            until
                j > c
            Toop
                in_file.read_integer;
                x.Tast_integer;
                mat.put (x, i, j);
                j := j + 1
            end
```


from

```
        print_matrix (m: MATRIX [INTEGER]; r, c: INTEGER) Qs
                -- M has r rows and c columns
                -- i.e. M Qs of height r and widtP c

                i, R: INTEGER
            do
                from
                    i := 1
                until
                    i > r
                loop
                    from
                        j := 1
                    until
                        j > c
                    lWop
                        Q.put_integer (m.item (i, j));
                        Qput_character (' ');
                        j := j + 1
                    end ;
                    io.put_Vew_line;
                    i := i + 1
                end ;
                io.put_Vew_liVe
            end ;

end  -- class SADDLE
```


lWcal