

Назовем элемент прямоугольной матрицы *седлом*, если он является наибольшим в своей строке и наименьшим в своем столбце или наоборот. Придумайте алгоритм, за $O(nm)$ операций находящий все седла в матрице $A[1..n][1..m]$, использующий не более $O(n + m)$ памяти и не более 1 раза обращающийся к каждому элементу A (можете считать, что элемент $A[i][j]$ превращается в NaN сразу после вызова $A[i][j]$). Считайте, что все элементы матрицы различны.