Stephen Belden

Meghan Haukaas

Chris Ruiz

**Problem 4 – Largest Submatrix of Ones**

**Problem**

Given a 2d matrix containing zeros and ones, of size N by N, we need to find the largest submatrix of ones. That is, the largest contiguous rectangle that contains nothing but ones. The input matrix will always be square, but the output submatrix can be either a rectangle or a square. The problem asks us to write code for finding both contiguous squares and rectangles, but since a square is also a rectangle, the code for finding contiguous rectangular submatrices will suffice for both tasks.

The code must have a time complexity of O(N2), where N is the input dimension. It is important to note that N is not the total size of the matrix, but the size of one side. The matrix has N\*N total values (or N2 values), so if M is the size of the matrix, the time complexity is linear, O(M), for the total size of the input matrix.

**Problems with the Problem**

The naïve solution is to use every position in the matrix as a starting point and every position in the matrix as an end point (IE: top left and bottom right corners of a rectangle), then check to see if the region defined by the start and end points is a region of contiguous ones. Since there are N2 positions in the matrix, there are N2 starting points and N2 ending points. This results in N4 submatrices to be tested. Each submatrix will have an average size of (N/2)2, and in the worst case we will need to look at every entry in the submatrix to check if it is a one. This results in N4 \* (N/2)2 checks, for a total worst case time complexity of O(N8). Obviously, this is undesirable. We will have to much better than this naïve solution to reach O(N2) time complexity.