

# INFSCI 2591: Algorithm Design

## Assignment 2

**Due: February 14, 2017**

- a. Write a pseudocode for the algorithm

**Problem:** Find the maximum width K of a square of 1's in A

**Input:** Matrix A[m][n] whose elements are either 1 or 0

**Output:** The indices (I, J) at the top left corner of the maximum square

```
int [] position(A[][]){  
    index i, j;  
    Array int [2] result;  
    Value maxWid = 0, temp;  
    for(i from (1 to m - 1)){  
        for(j from (1 to n - 1)){  
            if(A[i][j] == 1){  
                A[i][j] += min(A[i - 1][j - 1], A[i - 1][j], A[i][j - 1]);  
                if(A[i][j] > maxWid){  
                    result[0] = i;  
                    result[1] = j;  
                    maxWid = A[i][j];  
                }  
            }  
        }  
    }  
}
```

```
}
```

```
// this loop is to reset the matrix, which every element represents the maximum  
width of "1-square" regarding the element as right bottom corner
```

```
//former result stores the index of right bottom corner of the maximum square,  
then we need to compute the left top corner
```

```
result[0] = result[0] - maxWid + 1;
```

```
result[1] = result[1] - maxWid + 1;
```

```
return result;
```

```
// traverse matrix and find squares with maximum width when regarding each element  
as top left corner
```

- b. What is the time complexity of the algorithm?

$O(m*n)$