# ECE374 SP23 HW6

## Contributors

Zhirong Chen (zhirong4)

Ziyuan Chen (ziyuanc3)

## Problem 1

**Largest Square of $1$'s.** You are given a $n \times n$ bitonic array $A$ and the goal is to find the set of elements within that array that form a square filled with only $1$'s.

## Solution

*Memoization.* We define a function $LS(i, j), i, j \in [1, n+1]$. It records the size of the largest square of $1$'s that can be formed with the top-left corner at $(i, j)$. The matrix is filled from bottom-right to top-left.

*Recurrence function.* The function is intentionally defined to "overflow" at the bottom and right edges to simplify the recurrence.

$$LS(i,j) = \begin{cases} 0 & \begin{array}{r} A(i,j) = 0 \\ \text{or } i = n+1 \\ \text{or } j = n+1 \end{array} \\ 1 + \min \begin{Bmatrix} LS(i+1, j) \\ LS(i, j+1) \\ LS(i+1, j+1) \end{Bmatrix} & \text{otherwise} \end{cases}$$

The intuition is that if the current cell is $1$, then it can "merge" the squares of its neighbors to form a larger square; otherwise, it acts as a "obstacle" and forcefully resets the counter.

*Time Complexity.* Since we are filling $O(n^2)$ cells and each cell takes $O(1)$ time to compute, the time complexity is $O(n^2)$.