You have a 2D binary matrix that's filled with `0`s and `1`s. In the matrix, find the largest square that contains only `1`s and return its area.

**Example**

For

```
matrix = [
    ['1', '0', '1', '1', '1'],
    ['1', '0', '1', '1', '1'],
    ['1', '1', '1', '1', '1'],
    ['1', '0', '0', '1', '0'],
    ['1', '0', '0', '1', '0']
]
```

the output should be
`maximalSquare(matrix) = 9`.

**Input/Output**

- **[time limit] 4000ms (py3)**
- **[input] array.array.char matrix**

  *Guaranteed constraints:*
  `0 ≤ matrix.length ≤ 100`,
  `1 ≤ matrix[i].length ≤ 100`,
  `0 ≤ matrix[i][j] ≤ 1`.

- **[output] integer**

  An integer that represents the area of the largest square in the given matrix that is composed only of `1`s.