```c
#define ALLOC_LENGTH            (100000)

int largestRectangleArea(int* heights, int heightsSize){
    int max;
    int area;
    int* stack;
    int stack_ptr;
    int index;
    int cur_height;

    stack_ptr = -1;
    stack = (int*)malloc(sizeof(int)*ALLOC_LENGTH);
    max = 0;

    for(index = 0; index < (heightsSize+1); index++)
    {
        cur_height = (index == heightsSize) ? 0 : heights[index];

        if( (-1 == stack_ptr) || (cur_height > heights[stack[stack_ptr]]) )
        {
            stack_ptr++;
            stack[stack_ptr] = index;
        }else
        {
            area = stack[stack_ptr];
            stack_ptr--;
            area =  ((stack_ptr == -1) ? index : (index - stack[stack_ptr]-1) ) * heights[area];
            max = (area > max) ? area : max;

            index--;
        }

    }

    return max;

}

int maximalRectangle(char** matrix, int matrixSize, int* matrixColSize) {
    int* heights;
    int row_index;
    int col_index;
    int max;
    int value;

    heights = (int*)malloc(sizeof(int)*matrixColSize[0]);
    memset(heights, 0x0, sizeof(int)*matrixColSize[0]);
    max = 0;

    for(row_index = 0; row_index < matrixSize; row_index++)
    {
        for(col_index = 0; col_index < matrixColSize[0]; col_index++)
        {
            heights[col_index] = (matrix[row_index][col_index] == '0') ? 0 : heights[col_index]+1;
        }

        value = largestRectangleArea(heights, matrixColSize[0]);

        if(value > max)
        {
            max = value;
        }
    }

    return max;
}
```