Máximo rectángulo

La idea es buscar el rectágulo más grande formado solo por 0's. Inicialmente se podría pensar en una función iterativa, que en cada coordenada de la matriz calcule el rectangulo más grande. Pero estoy puede ser torpe y poco eficiente, por lo cual no es recomendable.

Consultando sobre el algoritmo de Kadene en 2D, fue notorio que una manera más eficiente de resolver el problema, es trasformandolo en otra matriz, en la cual la suma máxima de las submatrices, coincida con el rectángulo más grande formado por 0's. Para ello, se hace una matriz de penalización. En esta matriz, se convierten los 0's en 1's, y los 1's, son reemplazados con la constante $k = -n*m$.

En esta matriz, gracias a los 1's, el valor de la suma maś grande, coincide con el rectángulo más grande del problema inicial. La constante $k = -n * m$ asegura que la matriz no incluye 1's, ya que al suman este valor, la penalización es tan grande que el valor calculado es menor que 0, y por ende, no puede maximizar la suma.

Máximo Cuadrado

Aunque pareciera logico que al restringir la función calculada anteriormente, a un cuadrado, solucionaria directamente el problema, no se encontro una manera eficiente de restrigir la función a un cuadrado, por lo cual fue necesario buscar una solución alternativa.

Para esto, una solución inicial era mirar cual era la distancia más cercana de una celda con un 0, a otra celda con 1, pero esto es computacionalmente costoso. Siguiendo esta idea en mente, se puede trabajar a través de una matriz dinamica, que cambie una celda a 1, si esta tiene un 1 a derecha, abajo o en diagonal. Esto implicitamente calcula la distancia de una celda con un 0, al 1 más cercano que impide que sea un cuadrado más grande, por tanto, nos da el lado del cuadrado, que correspondera al número de iteraciones. Por último, el cuadrado más grande correspondera a la última celda con 0, reemplazada con un 1. Esto se puede evaluar mirando al final de una iteración si la matriz se convierte en la matriz de 1's de tamaño $(n * m)$, y cualquier elemento que en la iteración anterior fuera 0, corresponde a una posible solución.

Iterador de celdas libres

Para resolver este problema inicialmente se crea una función que determine si una celda es libre o no, segun la definición. Con esta función se crea un iterador, que arranque en la casilla indicada, y cada vez que encuentre una casilla adyacente, que sea tambien libre según la función creada inicialmente, la agregara al iterador. Este proceso se repite hasta que todos los elementos estan en el iterador. La respuesta final es son n coordenadas, todas desde las que se puede llegar con un camino de celdas libres iniciando desde la coordenada entregada a la función. Se tiene en cuenta el caso especial donde la celda de la matriz entregada resulte no ser libre. En este caso no existe camino alguno, o elementos en este camino, por tanto, se entrega la respuesta $'False'$