

Question 4

Zeal Liang

Z5325156

First, we expand the array A into a two-dimensional array and re-store the values in binary.

Then we create a two-dimensional array B of the same size as array A , and a linked list array C of length n .

Then we iterate through the array A .

When $A[i][j] = 1$, the chain of $C[i-1]$ is appended with i , $B[i][j] = C[i-1]$.

When $A[i][j] = 0$, $B[i][j] = B[i-1][j]$.

Finally, we traverse the array C and find the longest linked list. We use the elements in that list as indices to access the original array A , and we get the subarray we need.

Given that all our operations access $m \times n$ -sized two-dimensional arrays sequentially, the time complexity of the algorithm is $O(mn)$.