Maximum size square sub-matrix with all 1s

Given a binary matrix, find out the maximum size square sub-matrix with all 1s.

For example, consider the below binary matrix.

```
0  1  1  0  1
1  1  0  1  0
0  1  1  1  0
1  1  1  1  0
1  1  1  1  1
0  0  0  0  0
```

The maximum square sub-matrix with all set bits is

```
1  1  1
1  1  1
1  1  1
```

Algorithm:
Let the given binary matrix be M[R][C]. The idea of the algorithm is to construct an auxiliary size matrix S[][] in which each entry S[i][j] represents size of the square sub-matrix with all 1s including M[i][j] where M[i][j] is the rightmost and bottommost entry in sub-matrix.

```
1) Construct a sum matrix S[R][C] for the given M[R][C].
     a) Copy first row and first columns as it is from M[][] to S[][]
     b) For other entries, use following expressions to construct S[][]
         If M[i][j] is 1 then
            S[i][j] = min(S[i][j-1], S[i-1][j], S[i-1][j-1]) + 1
         Else /*If M[i][j] is 0*/
            S[i][j] = 0
2) Find the maximum entry in S[R][C]
3) Using the value and coordinates of maximum entry in S[i], print
   sub-matrix of M[][]
```

For the given M[R][C] in above example, constructed S[R][C] would be:

```
0  1  1  0  1
1  1  0  1  0
0  1  1  1  0
1  1  2  2  0
1  2  2  3  1
0  0  0  0  0
```

The value of maximum entry in above matrix is 3 and coordinates of the entry are (4, 3). Using the maximum value and its coordinates, we can find out the required sub-matrix.

```
#include<stdio.h>
#define bool int
#define R 6
#define C 5

void printMaxSubSquare(bool M[R][C])
{
  int i,j;
  int S[R][C];
  int max_of_s, max_i, max_j;

  /* Set first column of S[][]*/
  for(i = 0; i < R; i++)
     S[i][0] = M[i][0];

  /* Set first row of S[][]*/
  for(j = 0; j < C; j++)
     S[0][j] = M[0][j];

  /* Construct other entries of S[][]*/
  for(i = 1; i < R; i++)
  {
    for(j = 1; j < C; j++)
    {
```

```
      if(M[i][j] == 1)
        S[i][j] = min(S[i][j-1], S[i-1][j], S[i-1][j-1]) + 1;
      else
        S[i][j] = 0;
    }
  }

  /* Find the maximum entry, and indexes of maximum entry
     in S[][] */
  max_of_s = S[0][0]; max_i = 0; max_j = 0;
  for(i = 0; i < R; i++)
  {
    for(j = 0; j < C; j++)
    {
      if(max_of_s < S[i][j])
      {
         max_of_s = S[i][j];
         max_i = i;
         max_j = j;
      }
    }
  }

  printf("\n Maximum size sub-matrix is: \n");
  for(i = max_i; i > max_i - max_of_s; i--)
  {
    for(j = max_j; j > max_j - max_of_s; j--)
    {
      printf("%d ", M[i][j]);
    }
    printf("\n");
  }
}

/* UTILITY FUNCTIONS */
/* Function to get minimum of three values */
int min(int a, int b, int c)
{
  int m = a;
  if (m > b)
    m = b;
  if (m > c)
    m = c;
  return m;
}

/* Driver function to test above functions */
int main()
{
  bool M[R][C] =  {{0, 1, 1, 0, 1},
                   {1, 1, 0, 1, 0},
                   {0, 1, 1, 1, 0},
                   {1, 1, 1, 1, 0},
```

```
                    {1, 1, 1, 1, 1},
                    {0, 0, 0, 0, 0}};

  printMaxSubSquare(M);
  getchar();
}
```

Time Complexity: O(m*n) where m is number of rows and n is number of columns in the given matrix.
Auxiliary Space: O(m*n) where m is number of rows and n is number of columns in the given matrix.
Algorithmic Paradigm: Dynamic Programming

Please write comments if you find any bug in above code/algorithm, or find other ways to solve the same problem

Related Topics:

- Find Union and Intersection of two unsorted arrays
- Pythagorean Triplet in an array
- Maximum profit by buying and selling a share at most twice
- Design a data structure that supports insert, delete, search and getRandom in constant time
- Print missing elements that lie in range 0 – 99
- Iterative Merge Sort
- Group multiple occurrence of array elements ordered by first occurrence
- Given a sorted and rotated array, find if there is a pair with a given sum

Tags: Dynamic Programming

Writing code in comment? Please use **ideone.com** and share the link here.

Saurabh Shrivastava • 2 months ago

In my opinion this solution doesn't seem right .... correct me if I'm wrong.

The idea that you are taking minimum, somehow doesn't make sense, because say for M[i][j-1], there could be a case when all the number above are 1 and it will have a large value. So, i,j would be included in the "column" rectangle. Look at the below case

0 0 0 1 0
0 0 0 1 0
0 0 0 1 0
0 0 0 1 0
0 0 0 1 0
0 0 0 1 0

0 0 1 1 0
0 0 1 1 0
0 0 0 1 0

If you run you algorithm, you will end up on the small square because you are taking minimum. In my opinion, it should ideally be solved like this.

The S[R][C] will store the tuple (x, y) starting co ordinate of the enclosing rectangle.
So you S[i][j] = MAX(
(i - S[i-1][j].X) * (j - S[i-1][j].Y) ,
(i - S[i][j-1].X) * (j - S[i][j-1].Y) ,
(i - S[i][j-1].X) * (j - S[i-1][j].Y) // if S[i-1][j].Y > 2 & S[i][j-1].Y > 2
)

• Reply • Share ›

Anbarasan S • 2 months ago

I still don't get the logic of index (5,4) of the resulting matrix by applying the formula 1-b mentioned. Shouldn't it be 2 rather than 3(min(1,1,1)+1=2 ?

• Reply • Share ›

GOPI GOPINATH ➚ Anbarasan S • 2 months ago

Consider the new auxiliary array and not the original one. So it becomes min(2,2,2)+1 =. 3.

• Reply • Share ›

Jason Bourne • 4 months ago

I think the resultant matrix is not fully correct, it has to have value as 3 at position (4, 2), please correct if I am wrong

• Reply • Share ›

eknoor ➚ Jason Bourne • 3 months ago

No. Min(1,2,2)+1 =2

• Reply • Share ›

Jason Bourne ➚ eknoor • 2 months ago

got it, Thanks!

• Reply • Share ›

Anuj Garg • 4 months ago

GeeksforGeeks can be done in single traversal. No need for the second nested loop for finding max.

http://ideone.com/sSfI0k

• Reply • Share ›

Goku ➚ Anuj Garg • 4 months ago

Fails for this case :

{{0, 1, 1, 0, 1},

{1, 1, 0, 1, 0},

{0, 1, 1, 1, 0},

{1, 1, 1, 1, 0},

{1, 1, 1, 1, 0},

{0, 1, 1, 1, 0}}

Your code works only when maximum size sub matrix is a square matrix.

• Reply • Share ›

Goku ➚ Goku • 4 months ago

alright,its correct.We want square matrix only.

2 • Reply • Share ›

deeksha jain ➚ Goku • 2 months ago

Goku Your solution is correct right?
This algorithm anyways wont work for rectangular matrix.

• Reply • Share ›

Manraj Singh ➚ Anuj Garg • 4 months ago

Yeps.Adding to it, since it is finding the max side of square already and we know the square contains only 1 as elements you don't need to save max_i and max_j . Just use max_of_s to print the matrix.

Code:

Code:
http://ideone.com/6oHWCQ
Sorry for being cheap.Just saved two more variables. :P
• Reply • Share ›

Nitin Gupta • 5 months ago
Largest rectangle with all ones and Larges square with all ones
Along with printing rectangle and square...

"http://ideone.com/e.js/Y0wkXG"
• Reply • Share ›

Vijai • 6 months ago
Simple extension to rectangle is, take class point for (row,col). Use val at cur place is (min(compare row),min(compare col)).
• Reply • Share ›

The_Geek ➚ Vijai • 5 months ago
IMO, If u think deeply, this solution wont work out.
• Reply • Share ›

Vijai • 6 months ago
I thing, we should not include s[i-1][j-1] in finding minimum of 3 components...
• Reply • Share ›

tushar • 6 months ago
elegant..awesome
• Reply • Share ›

Siya • 6 months ago
Can any one tell me the intuition?How to think in such problem?
• Reply • Share ›

Ajay Gaur ➚ Siya • 4 months ago
Consider the auxiliary matrix and place aux[1][1], this is the minimum of all three squares(left, top and top-left) plus 1. if either of them is 0, then aux[1][1] will be 1 and we will come to know that there is only 1 matrix (of 1x1). if aux[1][1] = 2, then the minimum of those three would have been 1, that means all the 3 entries would have been 1, so there exists a 2x2 matrix whose all entries are 1. This can be used further inductively to find the maximum size submatrix.

Hope this helps.
2 • Reply • Share ›

dharam • 7 months ago

Here is the java implementation :-
https://github.com/dharam3/DS/...

• Reply • Share ›

Kenneth • 8 months ago

As the author's saying, we can use DP to get the maximum square area in the matrix. How about getting the maximum rectangle area in the matrix?
Here is my solution for rectangle area case by using increasing stack. Time complexity is O(m*n) and space complexity is O(m*n) .
http://ideone.com/3DmJEq

Tests:
0, 1, 1, 0, 1
1, 1, 1, 0, 0
0, 1, 1, 1, 0
1, 1, 1, 1, 0
1, 1, 1, 1, 1
0, 1, 1, 0, 0

The maximum rectangle area with all 1s is: 12

• Reply • Share ›

helper • 8 months ago

the simplest and clearest code following the above algo

ps: i figured out the algo myself :)

http://collabedit.com/ct384

• Reply • Share ›

ALEX • 8 months ago

check it ..... :)

http://ideone.com/nJnbUx

1 • Reply • Share ›

Shaifali Bhandari • 8 months ago

copied post from below explaining the rationale behind min (S[i][j-1], S[i-1][j], S[i-1][j-1])

Original:

1111
1110
1110
0111

Sum:

1110
1220
2230
0121

Explanation:

I am indexing from 0 for the sake of this explanation, with the left as the x-axis zero and the top as the y-axis zero.

see more

• Reply • Share ›

krishna • 8 months ago

different approach
http://ideone.com/KEKo67

• Reply • Share ›

Sherlock • 9 months ago

Recursion with Memoization:

```
/* initial call: maxOneSqMat(0, 0, rows, cols);*/

void maxOneSqMat(int r, int c, int rows, int cols)
{
if(res[r][c] == -1) {
if(r == 0)
res[0][c] = a[0][c];
else if(c == 0)
res[r][0] = a[r][0];
else {
if(a[r][c] == 1)
res[r][c] = min(res[r-1][c], res[r][c-1], res[r-1][c-1]) + 1;
else
res[r][c] = 0;
}

if(c == cols-1)
maxOneSqMat(r+1, 0, rows, cols);
else
maxOneSqMat(r, c+1, rows, cols);
}
}
```

• Reply • Share ›

Anurag Singh • 9 months ago

We can keep track of max values while constructing the size matrix. Looping on matrix after construction is not really needed.

1 • Reply • Share ›

anon Anurag Singh • 6 months ago

Same thing

• Reply • Share ›

Jun • 10 months ago

elegant solution

• Reply • Share ›

Vikas Malviya Jun • 9 months ago

The explanantion is not there, please explain a little about this algorithm.Thanx

• Reply • Share ›

Vãîbhåv Joshî • 10 months ago

Can we further Optimize this algorithm O(m*n) to O(m+n)....?

• Reply • Share ›

rishabhjoshi • 10 months ago

there is a comment in reply ti gfg by max 3 years ago he has explained it really well!!

• Reply • Share ›

Paparao Veeragandham • 10 months ago

```
int MaxSizeRect(int data[row][col] , int row , int col)
{
//out Max size in Rectangle 2D Matrix.
//temp1 for row wise bigger. temp2 col wise bigger for Matrix.
int temp1[row][col] , temp2[row][col];
int max = INT_MIN;

for ( i = 0; i < row; i++)
{
temp1[i][0] = data[i][0];
temp2[i][0] = data[i][0];
}
for( i = 0 ; i< col; i++)
{
temp1[i][0] = data[i][0];
temp2[i][0] = data[i][0];
```

}

see more

• Reply • Share ›

Bottom Boy • a year ago

First you need to divided the problems in sub problems.
Then combine the solution.
Using this approach the complexity of this problem will be more.

• Reply • Share ›

epsIon • a year ago

Can anyone explain how to approach such questions?

2 • Reply • Share ›

vpr ➚ epsIon • a year ago

below there's comment by gksgeek (posted 3 yrs ago)
in that conversation, Max tried to explain the concept... see if that helps.

• Reply • Share ›

wrestler • a year ago

Index of max is not needed, simply print the square with all 1 of size max.

5 • Reply • Share ›

ANA ➚ wrestler • a year ago

how ?

• Reply • Share ›

Anurag Singh ➚ ANA • 9 months ago

from i = 1 to max
------for j = 1 to max
----------printf("1 ");
------printf("\n")

1 • Reply • Share ›

Meenal Mishra • a year ago

Guys check this should be helpful
http://www.queryhome.com/26246...

• Reply • Share ›

Bottom Boy • a year ago

typedef struct mat

```
{
int n, r, c;
}matParm;

matParm sub_mat; // Global Variable
/* Call from main function */

if( row > col)
for (i = 0; i < row - col; i++)
find_sqr_mat (matrix, i, 0, col);
else if (col > row)
for (i = 0; i < col - row ; i++)
find_sqr_mat (matrix, 0, i, row);
else
find_sqr_mat (matrix, 0, 0, col);
/* Print the out put matrix */

printf("\n Largest Sub Square Matrix\n");
```

see more

• Reply • Share ›

prashant jha • a year ago

my c++ code using dp in o(mn) compleity
http://ideone.com/anhb8j

• Reply • Share ›

Nikhil Kumar • a year ago

can any one explain the logic of above algorithm ?
why we take min of three values and add 1 if M[i][j] == 1 ?

7 • Reply • Share ›

Timo ➚ Nikhil Kumar • 9 months ago

As ANA said, visualizing squares is key to understand why the minimum over all neighboring elements is chosen.

Consider the case for an element in S to become 2 (i.e., a 2 by 2 square matrix). You'd need its neighbors at the top, left, and top-left to be trivial 1 by 1 square matrices, which is the same as the respective values in S to be 1 each.

Now, for a 3 times 3 square matrix, you'd need each such neighbor to be a 2 times 2 square matrix. This happens to be equal to the respective values in S be 2 each. (Imagine 2 by 2-sized rectangle borders drawn around the neighboring squares to help understand.) This continues for larger squares similarly.

If any of the neighboring squares happens to be smaller than the other neighbors, you

cannot have an extended, larger square (because such a smaller neighboring square is basically "breaking" the perfect larger square you could get otherwise.) What's the maximum square you can get in this case? It's the one defined by the smallest neighboring square. This turns out to be defined by the minimum over all neighboring squares.

Hope this helps!

4 ^ | v • Reply • Share ›

ANA ➚ Nikhil Kumar • a year ago

If M[i][j] is '1' then it will contribute to the all 1s square sub-matrix ending at either M[i][j-1] or M[i-1][j] or M[i-1][j-1]. If we visualize the conditions then, we will see:
S[i][j] = min(S[i][j-1], S[i-1][j], S[i-1][j-1]) + 1

^ | v • Reply • Share ›

Sachin • a year ago

I think we can reduce the space complexity to m x 2(m-no. of rows); since the value only depends on 3 cells, and can store the maximum value in a separate variable, to keep the track of biggest square sub matrix. This will drastically reduce the space complexity to O(2*m) ~ O(m)

1 ^ | v • Reply • Share ›

Neha Garg • 2 years ago

complexity can be reduced keep the track of maxofs in the first step .. then no need to again track the s matrix for finding it and also no need to max_i and max_j as all elements are 1 and we dont want position of that matrix .. the maxofs will give dimension of required matrix as this is a square matrix
plz correct me if i m wrong...

^ | v • Reply • Share ›

draganwarrior • 2 years ago

solution for rectangle http://ideone.com/DbafRI

^ | v • Reply • Share ›

vishal ➚ draganwarrior • 2 years ago

your soln doesnt work for the given input:
{{1,1,1,1,0,0,1,0,1,1},
{1,0,1,0,1,1,0,1,0,1},
{1,1,1,1,1,1,0,0,0,0},
{0,1,1,1,1,1,1,1,1,0},
{1,1,1,1,0,1,0,1,0,1}}
output will be
1 1 1 1 1
1 1 1 1 1

but your output is {{1},{1}}

• Reply • Share ›

Load more comments

Popular Posts

- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

- Follow @GeeksforGeeks

Recent Comments

- lt_k

  i need help for coding this function in java...

  Java Programming Language · 1 hour ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 1 hour ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 1 hour ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

  Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 3 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 3 hours ago