Cross Matrix

You are given a *N* * *N* matrix, *U*. You have to choose 2 sub-matrices A and B made of only 1s of *U*, such that, they have at least 1 cell in common, and each matrix is not completely engulfed by the other i.e.,

If *U* is of the form

$$U = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-2} & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-2} & a_{2,N-1} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{N-1,0} & a_{N-1,1} & \cdots & a_{N-1,N-2} & a_{N-1,N-1} \end{bmatrix}$$

and *A* is of the form

$$A = \begin{bmatrix} a_{x_1,y_1} & \cdots & a_{x_1,y_2} \\ . & . & . \\ . & . & . \\ . & . & . \\ a_{x_2,y_1} & \cdots & a_{x_2,y_2} \end{bmatrix}$$

and *B* is of the form

$$B = \begin{bmatrix} a_{x_3,y_3} & \cdots & a_{x_3,y_4} \\ . & . & . \\ . & . & . \\ . & . & . \\ a_{x_4,y_3} & \cdots & a_{x_4,y_4} \end{bmatrix}$$

then, there exists atleast 1 $a_{i,j} : a_{i,j} \in A$ and $a_{i,j} \in B$
then, there exists atleast 1 $a_{i1,j1} : a_{i1,j1} \in A$ and $a_{i1,j1} \notin B$
then, there exists atleast 1 $a_{i2,j2} : a_{i2,j2} \in B$ and $a_{i2,j2} \notin A$
$a_{x,y} = 1 \; \forall \; a_{x,y} \in A$
$a_{x,y} = 1 \; \forall \; a_{x,y} \in B$

How many such (*A*, *B*) exist?

Input Format

The first line of the input contains a number *N*.
N lines follow, each line containing *N* integers (0/1) NOT separated by any space.

Output Format

Output the total number of such (A, B) pairs. If the answer is greater than or equal to $10^9 + 7$, then print answer modulo (%) $10^9 + 7$.

Constraints

$2 \leq N \leq 1500$
$a_{i,j} \in [0, 1] : 0 \leq i, j \leq N - 1$

Sample Input

```
4
0010
0001
1010
1110
```

Sample Output

```
10
```

Explanation

X means the common part of A and B.
We can swap A and B to get another answer.

```
0010
0001
A010
XB10

0010
0001
A010
XBB0

0010
0001
10A0
1BX0

0010
0001
10A0
BBX0

0010
0001
1010
AXB0
```

TimeLimits

Time limit for this challenge is mentioned here

Contest ends in a day

Submissions: 23
Max Score: 75

Rate This Challenge:

More

C++

```
1 #include <cstdio>
2 #include <cstring>
3 #include <string>
4 #include <iostream>
5 #include <sstream>
```

```cpp
#include <map>
#include <set>
#include <vector>
#include <queue>
#include <bitset>
#include <numeric>
#include <ctime>
#include <cmath>
#include <cassert>
#include <algorithm>

using namespace std;

typedef pair<int, int> PII;
typedef long long ll;

#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define pct __builtin_popcount

#define N 1510
#define P 1000000007
#define I36 27777778

int add(int a, int b) {
    return (a+b)%P;
}

int sub(int a, int b) {
    return (a-b+P)%P;
}

int mul(int a, int b) {
    return (ll)a*b%P;
}

struct Poly {
    int A,B,C;
    // Ax^2+Bx+C
    Poly() {A = B = C = 0;}
    Poly(int _A, int _B, int _C):A(_A), B(_B), C(_C) {}

    Poly operator + (Poly a) {
        return Poly(add(A,a.A),add(B,a.B),add(C,a.C));
    }

    Poly operator - (Poly a) {
        return Poly(sub(A,a.A),sub(B,a.B),sub(C,a.C));
    }

    Poly operator * (int a) {
        return Poly(mul(A,a),mul(B,a),mul(C,a));
    }

    int eval(int x) {
        return add(add(mul(mul(A,x),x),mul(B,x)),C);
    }
};

Poly pol(int L, int R, int H) {
    Poly p;
    p.A = 3;
    p.B = sub(sub(6,mul(3,L)),mul(3,R));
    p.C = add(add(sub(add(sub(2,mul(3,L)),mul(L,L)),mul(3,R)),mul(L,R)),mul(R,R));
    p = p*sub(L,R);
    p = p*H;
    p = p*add(1,H);
    p = p*add(2,H);
    p = p*I36;
```

```
    return p;
}

int n;
string s[N];
int h[N];

int a[N][N];
int st[N], sh[N];
int b[N][N], c[N][N], d[N][N];

string rots[N];
void rot() {
    for (int i = 0; i < n; i ++)
        rots[i] = s[i];
    for (int i = 0; i < n; i ++)  {
        for (int j = 0; j < n; j ++) {
            s[n-j-1][i] = rots[i][j];
        }
    }
}

int roti[N][N];
void irot(int a[][N]) {
    for (int i = 0; i < n; i ++) {
        for (int j = 0; j < n; j ++)
            roti[i][j] = a[i][j];
    }
    for (int i = 0; i < n; i ++) {
        for (int j = 0; j < n; j ++) {
            a[i][j] = roti[n-j-1][i];
        }
    }
}

void calc(int a[][N]) {
    memset(h,0,sizeof h);
    for (int i = 0; i < n; i ++) {
        for (int j = 0; j < n; j ++) {
            if (s[i][j] == '1') h[j] ++;
            else h[j] = 0;
        }
        int sn = 0;
        int uS = 0;
        st[0] = -1;
        sh[0] = 0;
        sn ++;
        for (int j = 0; j < n; j ++) {
            while (sn != 1 && sh[sn-1] > h[j]) {
                uS = sub(uS, mul(sub(st[sn-1],st[sn-2]),sh[sn-1]));
                sn --;
            }
            st[sn] = j;
            sh[sn] = h[j];
            sn ++;
            uS = add(uS, mul(sub(st[sn-1],st[sn-2]),sh[sn-1]));
            a[i][j] = uS;
        }
    }
}

void print(int a[][N]) {
    return;
    for (int i = 0; i < n; i ++) {
        for (int j = 0; j < n; j ++)
            printf ("%d%c",a[i][j],j==n-1?'\n':' ');
    }
    puts ("");
}

int main()
```

```
{
    cin >> n;
    for (int i = 0; i < n; i ++)
        cin >> s[i];
    int S = 0;
    int T = 0;
    for (int i = 0; i < n; i ++) {
        for (int j = 0; j < n; j ++) {
            if (s[i][j] == '1') h[j] ++;
            else h[j] = 0;
        }
        int sn = 0;
        int uS = 0;
        Poly p;
        st[0] = -1;
        sh[0] = 0;
        sn ++;
        for (int j = 0; j < n; j ++) {
            while (sn != 1 && sh[sn-1] > h[j]) {
                uS = sub(uS, mul(sub(st[sn-1],st[sn-2]),sh[sn-1]));
                p = p-pol(st[sn-1],st[sn-2],sh[sn-1]);
                sn --;
            }
            st[sn] = j;
            sh[sn] = h[j];
            sn ++;
            uS = add(uS, mul(sub(st[sn-1],st[sn-2]),sh[sn-1]));
            p = p+pol(st[sn-1],st[sn-2],sh[sn-1]);
            S = add(S, uS);
            T = add(T, p.eval(j));
            a[i][j] = uS;
        }
    }

    rot(); calc(b); irot(b);
    rot(); calc(c); irot(c); irot(c);
    rot(); calc(d); irot(d); irot(d); irot(d);

    print(a);
    print(b);
    print(c);
    print(d);

    int TT = 0;
    for (int i = 0; i < n; i ++) {
        for (int j = n-1; j >= 0; j --) {
            if (i > 0) b[i][j] = add(b[i][j],b[i-1][j]);
            if (j < n-1) b[i][j] = add(b[i][j],b[i][j+1]);
            if (i > 0 && j < n-1) b[i][j] = sub(b[i][j],b[i-1][j+1]);
        }
    }

    for (int i = n-1; i >= 0; i --) {
        for (int j = n-1; j >= 0; j --) {
            if (i < n-1) c[i][j] = add(c[i][j],c[i+1][j]);
            if (j < n-1) c[i][j] = add(c[i][j],c[i][j+1]);
            if (i < n-1 && j < n-1) c[i][j] = sub(c[i][j],c[i+1][j+1]);
        }
    }

    print(b);
    print(c);

    for (int j = 0; j < n-1; j ++) {
        int U = 0;
        for (int i = 0; i < n; i ++) {
            U = add(U, a[i][j]);
        }
        TT = add(TT, mul(U, c[0][j+1]));
    }

```

```
    for (int i = 0; i < n-1; i ++) {
        int U = 0;
        for (int j = 0; j < n; j ++) {
            U = add(U, a[i][j]);
        }
        TT = add(TT, mul(U, c[i+1][0]));
    }

    for (int i = 0; i < n-1; i ++) {
        for (int j = 0; j < n-1; j ++) {
            TT = sub(TT, mul(a[i][j], c[i+1][j+1]));
        }
    }

    for (int i = 1; i < n; i ++) {
        for (int j = 0; j < n-1; j ++) {
            TT = sub(TT, mul(d[i][j], b[i-1][j+1]));
        }
    }

    int SS = 0;
    SS = add(SS, mul(S,S));
    SS = sub(SS, mul(2,T));
    SS = add(SS, S);
    SS = sub(SS, mul(2,TT));
    cout << SS << endl;
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code