

PREPARE^{NEW}

CERTIFY

COMPLETE

Search



TheTeam404

All Contests > SLIIT Codefest 2022 Hackathon - First round > Cross Matrix

Cross Matrix

Problem

Submissions

Leaderboard

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. Solved: 51 Attempted: 53 e.,

If U is of the form

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

and A is of the form

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

and B is of the form

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

then, there exists atleast 1 $a_{i,j} : a_{i,j} \in A$ and $a_{i,j} \in B$

then, there exists atleast 1 $a_{i_1,j_1} : a_{i_1,j_1} \in A$ and $a_{i_1,j_1} \notin B$

then, there exists atleast 1 $a_{i_2,j_2} : a_{i_2,j_2} \in B$ and $a_{i_2,j_2} \notin A$

$a_{x,y} = 1 \forall a_{x,y} \in A$

$a_{x,y} = 1 \forall a_{x,y} \in B$

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](#)

f t in

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>
```

```

6 #include <map>
7 #include <set>
8 #include <vector>
9 #include <queue>
10 #include <bitset>
11 #include <numeric>
12 #include <ctime>
13 #include <cmath>
14 #include <cassert>
15 #include <algorithm>
16
17 using namespace std;
18
19 typedef pair<int, int> PII;
20 typedef long long ll;
21
22 #define fi first
23 #define se second
24 #define mp make_pair
25 #define pb push_back
26 #define pct __builtin_popcount
27
28 #define N 1510
29 #define P 1000000007
30 #define I36 27777778
31
32 int add(int a, int b) {
33     return (a+b)%P;
34 }
35
36 int sub(int a, int b) {
37     return (a-b+P)%P;
38 }
39
40 int mul(int a, int b) {
41     return (ll)a*b%P;
42 }
43
44 struct Poly {
45     int A,B,C;
46     // Ax^2+Bx+C
47     Poly() {A = B = C = 0;}
48     Poly(int _A, int _B, int _C):A(_A), B(_B), C(_C) {}
49
50     Poly operator + (Poly a) {
51         return Poly(add(A,a.A),add(B,a.B),add(C,a.C));
52     }
53
54     Poly operator - (Poly a) {
55         return Poly(sub(A,a.A),sub(B,a.B),sub(C,a.C));
56     }
57
58     Poly operator * (int a) {
59         return Poly(mul(A,a),mul(B,a),mul(C,a));
60     }
61
62     int eval(int x) {
63         return add(add(mul(mul(A,x),x),mul(B,x)),C);
64     }
65 };
66
67 Poly pol(int L, int R, int H) {
68     Poly p;
69     p.A = 3;
70     p.B = sub(sub(6,mul(3,L)),mul(3,R));
71     p.C = add(add(sub(add(sub(2,mul(3,L)),mul(L,L)),mul(3,R)),mul(L,R)),mul(R,R));
72     p = p*sub(L,R);
73     p = p*H;
74     p = p*add(1,H);
75     p = p*add(2,H);
76     p = p*I36;

```

```

77     return p;
78 }
79
80 int n;
81 string s[N];
82 int h[N];
83
84 int a[N][N];
85 int st[N], sh[N];
86 int b[N][N], c[N][N], d[N][N];
87
88 string rots[N];
89 void rot() {
90     for (int i = 0; i < n; i++)
91         rots[i] = s[i];
92     for (int i = 0; i < n; i++) {
93         for (int j = 0; j < n; j++) {
94             s[n-j-1][i] = rots[i][j];
95         }
96     }
97 }
98
99 int roti[N][N];
100 void irot(int a[][N]) {
101     for (int i = 0; i < n; i++) {
102         for (int j = 0; j < n; j++)
103             roti[i][j] = a[i][j];
104     }
105     for (int i = 0; i < n; i++) {
106         for (int j = 0; j < n; j++) {
107             a[i][j] = roti[n-j-1][i];
108         }
109     }
110 }
111
112 void calc(int a[][N]) {
113     memset(h,0,sizeof h);
114     for (int i = 0; i < n; i++) {
115         for (int j = 0; j < n; j++) {
116             if (s[i][j] == '1') h[j]++;
117             else h[j] = 0;
118         }
119         int sn = 0;
120         int uS = 0;
121         st[0] = -1;
122         sh[0] = 0;
123         sn++;
124         for (int j = 0; j < n; j++) {
125             while (sn != 1 && sh[sn-1] > h[j]) {
126                 uS = sub(uS, mul(sub(st[sn-1],st[sn-2]),sh[sn-1]));
127                 sn--;
128             }
129             st[sn] = j;
130             sh[sn] = h[j];
131             sn++;
132             uS = add(uS, mul(sub(st[sn-1],st[sn-2]),sh[sn-1]));
133             a[i][j] = uS;
134         }
135     }
136 }
137
138 void print(int a[][N]) {
139     return;
140     for (int i = 0; i < n; i++) {
141         for (int j = 0; j < n; j++)
142             printf ("%d%c",a[i][j],j==n-1?'\\n':' ');
143     }
144     puts ("");
145 }
146
147 int main()

```

```

148 {
149     cin >> n;
150     for (int i = 0; i < n; i++)
151         cin >> s[i];
152     int S = 0;
153     int T = 0;
154     for (int i = 0; i < n; i++) {
155         for (int j = 0; j < n; j++) {
156             if (s[i][j] == '1') h[j]++;
157             else h[j] = 0;
158         }
159         int sn = 0;
160         int uS = 0;
161         Poly p;
162         st[0] = -1;
163         sh[0] = 0;
164         sn++;
165         for (int j = 0; j < n; j++) {
166             while (sn != 1 && sh[sn-1] > h[j]) {
167                 uS = sub(uS, mul(sub(st[sn-1], st[sn-2]), sh[sn-1]));
168                 p = p-pol(st[sn-1], st[sn-2], sh[sn-1]);
169                 sn--;
170             }
171             st[sn] = j;
172             sh[sn] = h[j];
173             sn++;
174             uS = add(uS, mul(sub(st[sn-1], st[sn-2]), sh[sn-1]));
175             p = p+pol(st[sn-1], st[sn-2], sh[sn-1]);
176             S = add(S, uS);
177             T = add(T, p.eval(j));
178             a[i][j] = uS;
179         }
180     }
181
182     rot(); calc(b); irot(b);
183     rot(); calc(c); irot(c); irot(c);
184     rot(); calc(d); irot(d); irot(d); irot(d);
185
186     print(a);
187     print(b);
188     print(c);
189     print(d);
190
191     int TT = 0;
192     for (int i = 0; i < n; i++) {
193         for (int j = n-1; j >= 0; j--) {
194             if (i > 0) b[i][j] = add(b[i][j], b[i-1][j]);
195             if (j < n-1) b[i][j] = add(b[i][j], b[i][j+1]);
196             if (i > 0 && j < n-1) b[i][j] = sub(b[i][j], b[i-1][j+1]);
197         }
198     }
199
200     for (int i = n-1; i >= 0; i--) {
201         for (int j = n-1; j >= 0; j--) {
202             if (i < n-1) c[i][j] = add(c[i][j], c[i+1][j]);
203             if (j < n-1) c[i][j] = add(c[i][j], c[i][j+1]);
204             if (i < n-1 && j < n-1) c[i][j] = sub(c[i][j], c[i+1][j+1]);
205         }
206     }
207
208     print(b);
209     print(c);
210
211     for (int j = 0; j < n-1; j++) {
212         int U = 0;
213         for (int i = 0; i < n; i++) {
214             U = add(U, a[i][j]);
215         }
216         TT = add(TT, mul(U, c[0][j+1]));
217     }
218 }

```

```
219  for (int i = 0; i < n-1; i++) {
220      int U = 0;
221      for (int j = 0; j < n; j++) {
222          U = add(U, a[i][j]);
223      }
224      TT = add(TT, mul(U, c[i+1][0]));
225  }
226
227  for (int i = 0; i < n-1; i++) {
228      for (int j = 0; j < n-1; j++) {
229          TT = sub(TT, mul(a[i][j], c[i+1][j+1]));
230      }
231  }
232
233  for (int i = 1; i < n; i++) {
234      for (int j = 0; j < n-1; j++) {
235          TT = sub(TT, mul(d[i][j], b[i-1][j+1]));
236      }
237  }
238
239  int SS = 0;
240  SS = add(SS, mul(S,S));
241  SS = sub(SS, mul(2,T));
242  SS = add(SS, S);
243  SS = sub(SS, mul(2,TT));
244  cout << SS << endl;
245  return 0;
246 }
```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)[Run Code](#)[Submit Code](#)