



שאלה 3 (25 נקודות)

נתונה מטריצה a בגודל N על N , המכילה מספרים שלמים. לכל איבר $a[i][j]$ במטריצה זו, נגדיר את ארבע תת המטריצות הבאות שאיבר זה מהווה פינה שלהם:

תת-מטריצה שמאלית עליונה: האיברים $a[0..i][0..j]$
 ימנית עליונה: האיברים $a[0..i][j..N-1]$
 שמאלית תחתונה: האיברים $a[i..N-1][0..j]$
 ימנית תחתונה: האיברים $a[i..N-1][j..N-1]$

לדוגמה, עבור המטריצה הבאה ($N=4$),

2	1	2	2
6	1	2	2
5	1	2	3
8	2	7	1

האיבר $a[2][1]$ (המסומן באפור) מגדיר את ארבע תת-המטריצות:

<table border="1"> <tr><td>2</td><td>1</td></tr> <tr><td>6</td><td>1</td></tr> <tr><td>5</td><td>1</td></tr> </table>	2	1	6	1	5	1	<table border="1"> <tr><td>1</td><td>2</td><td>2</td></tr> <tr><td>1</td><td>2</td><td>2</td></tr> <tr><td>1</td><td>2</td><td>3</td></tr> </table>	1	2	2	1	2	2	1	2	3
2	1															
6	1															
5	1															
1	2	2														
1	2	2														
1	2	3														
<table border="1"> <tr><td>5</td><td>1</td></tr> <tr><td>8</td><td>2</td></tr> </table>	5	1	8	2	<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>2</td><td>7</td><td>1</td></tr> </table>	1	2	3	2	7	1					
5	1															
8	2															
1	2	3														
2	7	1														

אנו נאמר כי איבר $a[i][j]$ במטריצה **מאוזן** אם סכום האיברים בכל אחת מארבע תת המטריצות שהוא מגדיר זהה. למשל, בדוגמה למעלה $a[2][1]$ מאוזן, כיוון שסכום כל אחת מארבע תת המטריצות שהוא מגדיר שווה ל-16: שימו לב שעבור תת-המטריצה השמאלית העליונה מתקיים $1+1+1+5+6+2=16$, עבור תת המטריצה השמאלית התחתונה מתקיים $2+8+5+1=16$, וכך הלאה.

עליכם לממש (בדף הבא) פונקציה שמקבלת מטריצה המיוצגת כמערך דו-ממדי $a[N][N]$ (מוגדר כ-`#define`). הפונקציה מחזירה 1 אם קיים במטריצה איבר מאוזן, ו-0 אחרת.

על הפונקציה לעבוד בסיבוכיות זמן $O(N^2)$ וסיבוכיות מקום $O(N^2)$. שימו לב: פתרונות בסיבוכיות גרועה מזו יזכו בניקוד מופחת, בהתאם לסיבוכיות הזמן של הפתרון.



```
int equal_submatrix_sum(int a[N][N]) {  
  
    int i,j;  
    int ul[N][N] = { 0 };  
  
    init_ul_matrix(a, ul);  
  
    for (i=0; i<N; ++i) {  
        for (j=0; j<N; ++j) {  
  
            // sums of upper-left, upper-right, lower-left and  
            // lower-right sub-matrices  
            int ul_sum, ur_sum, ll_sum, lr_sum;  
  
            ul_sum = ul[i][j];  
            ur_sum = ul[i][N-1] - (j>0 ? ul[i][j-1] : 0);  
            ll_sum = ul[N-1][j] - (i>0 ? ul[i-1][j] : 0);  
            lr_sum = ul[N-1][N-1] - (i>0 ? ul[i-1][N-1] : 0)  
                - (j>0 ? ul[N-1][j-1] : 0)  
                + (i>0 && j>0 ? ul[i-1][j-1] : 0);  
  
            if ( ul_sum == ur_sum && ur_sum == ll_sum &&  
                ll_sum == lr_sum )  
                return 1;  
        }  
    }  
    return 0;  
}
```



```
void init_ul_matrix(int a[N][N], int ul[N][N])
{
    int i,j;

    for (i=0; i<N; ++i) {
        for (j=0; j<N; ++j) {
            ul[i][j] = a[i][j];

            if (i>0)
                ul[i][j] += ul[i-1][j];
            if (j>0)
                ul[i][j] += ul[i][j-1];
            if (i>0 && j>0)
                ul[i][j] -= ul[i-1][j-1];
        }
    }
}
```

הסבר: המערך $ul[]$ מאוחל להכיל את הסכומים של תת-המטריצות השמאליות-עליונות:

$$ul[i][j] = \text{sum} \{ a[0..i][0..j] \}$$

מערך זה ניתן לחישוב בזמן $O(N^2)$, כיוון שלכל j , מתקיים השיויון

$$ul[i][j] = a[i][j] + ul[i-1][j] + ul[i][j-1] - ul[i-1][j-1]$$

ולכן חישוב כל תא דורש מספר קבוע של פעולות.

מתוך מערך העזר $ul[]$, ניתן לחשב ביעילות לכל תא i,j ב- a את סכום ארבע תת-המטריצות שסביבו.

לדוגמה, סכום תת המטריצה השמאלית-תחתונה מתקבל באמצעות הנוסחה

$$\text{sum} \{ a[i..N-1][0..j] \} = ul[N-1][j] - ul[i-1][j]$$

וכן הלאה.