



שאלה 2 (20 נקודות)

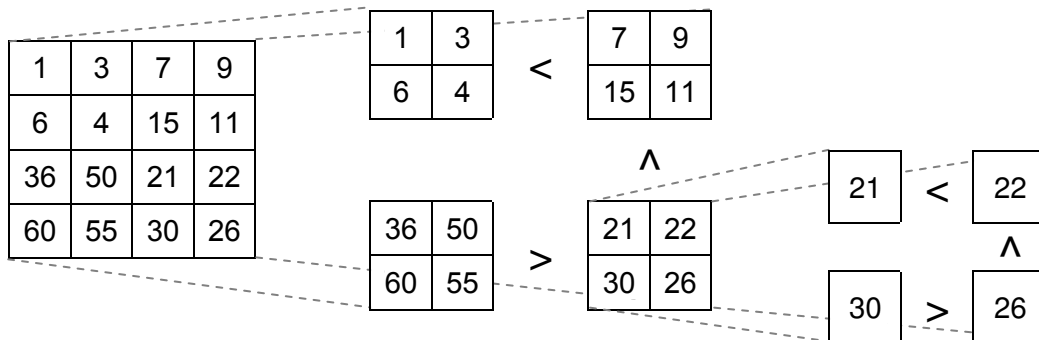
בשאלה זו נתייחס למערכים דו-ממדיים **ריבועיים** בגודל $N \times N$ (כאשר N הוא קבוע המוגדר כ-`#define`). לצורך השאלה, נניח כי N הוא חזקה שלמה של 2. עבור מערך כזה, נגדיר חלוקה פנימית שלו לארבעה רובעים בגודל $N/2 \times N/2$, ממוספרים מ-1 עד 4, באופן הבא:

1	2
4	3

נאמר שהמערך הוא **סיבובי** אם כל האיברים ברובע 1 קטנים ממש מכל אלו שברובע 2, אלו שברובע 2 קטנים ממש מכל אלו שברובע 3, ואלו שברובע 3 קטנים ממש מכל אלו שברובע 4. למשל, המערך הבא הוא סיבובי:

1	5
9	7

לשם הנוחות, נגדיר גם כל מערך בגודל 1×1 כמעריך סיבובי. כעת, נאמר שמערך $N \times N$ הוא **ממוין-סיבובי** אם הוא סיבובי, ארבעת הרובעים שלו סיבוביים, וכן הלאה עד לרובעים בגודל 1×1 . לדוגמה, המערך הבא ממוין-סיבובי:



עליכם לממש פונקציה לחיפוש במערך ממוין-סיבובי (בעמוד הבא). הפונקציה מקבלת כפרמטר את המערך הדו-ממדי $a[N][N]$, ואת הערך לחיפוש x . במידה ו- x נמצא במערך, הפונקציה תחזיר 1 ותכתוב את הקואורדינטות שלו למשתנים i, j הניתנים כפרמטרים. במידה ו- x אינו במערך, הפונקציה תחזיר 0 ואין חשיבות לתוכן של i, j .

דרישות סיבוכיות: על הפונקציה לעבוד בסיבוכיות זמן טובה ככל הניתן. פתרון לא אופטימאלי יזכה בניקוד חלקי לכל היותר. כמו כן על הפונקציה לעבוד בסיבוכיות מקום נוסף $O(1)$.



```
int find2d(int a[N][N], int x, int* i, int* j) {  
  
    int top=0, left=0, size=N;  
  
    while (size > 1) {  
        if (x > a[top + size/2 - 1][left]) {  
            // not in 1st quarter  
            if (x <= a[top + size/2 - 1][left + size/2]) {  
                // 2nd quarter  
                left += size/2;  
            } else if (x <= a[top + size - 1][left + size/2]) {  
                // 3rd quarter  
                top += size/2;  
                left += size/2;  
            } else  
                // 4th quarter  
                top += size/2;  
        }  
        size /= 2;  
    }  
  
    if (a[top][left] == x) {  
        *i = top;  
        *j = left;  
        return 1;  
    } else  
        return 0;  
}
```