



הנדסת תוכנה  
Software Engineering

תרגיל 2 להגשה במבני נתונים (קורס מס' 10117)

מרצים: ד"ר ראובן חוטובלי וד"ר דוד שטטר

תאריך הגשה: 22.12.2020 העבודה - בזוגות.

חובה לענות על כל השאלות ולתת את כל התשובות על גבי גיליון התשובות בלבד.

שאלה 1

לפניך 13 סעיפים שאינם תלויים זה בזה. ענה על כל הסעיפים. בכל סעיף יש היגד ועליכם לקבוע אם ההיגד נכון או לא. בעבור כל סעיף סמנו בגיליון התשובות ב-X במשבצת המתאימה - את תשובתכם. בתשובותיכם נא להתבסס על ההגדרות לפי גבולות.

א.  $7n^2 \in O(n^2)$

ב.  $7n^2 \in \Omega(n^2)$

ג.  $7n^2 \in \Theta(n^2)$

ד.  $7n^2 \in o(n^2)$  (או הקטן)

ה.  $7n \in O(n^2)$

ו.  $7n \in \Omega(n^2)$

ז.  $7n \in \Theta(n^2)$

ח.  $7n \in o(n^2)$

ט.  $7n^3 \in O(n^2)$

י.  $7n^3 \in \Omega(n^2)$

יא.  $7n^3 \in \Theta(n^2)$

יב.  $7n^3 \in o(n^2)$

יג.  $n \cdot 2^n \in O(3^n)$

שאלה 2

לפניך עשרים ושניים סעיפים שאינם תלויים זה בזה. ענה על כל הסעיפים. בכל סעיף נתונות ארבע תשובות, שרק אחת מהן נכונה. בכל סעיף, בחר את התשובה הנכונה וסמן אותה ב-X במשבצת המתאימה בגיליון התשובות.



א.

נתונה פונקציית זמן הריצה של אלגוריתם מסוים, הפועל על קלט שגודלו  $n$  :  
 $T(n) = 16T(n/2) + 2n^4$  . מהי סיבוכיות זמן הריצה של האלגוריתם?

1.  $\Theta(n^4)$  3.  $\Theta(n^2 \log n)$

2.  $\Theta(n^8 \log n)$  4.  $\Theta(n^4 \log n)$

ב.

נתונה פונקציית זמן הריצה של אלגוריתם מסוים, הפועל על קלט שגודלו  $n$  :  
 $T(n) = 2T(n/2) + n \log^2 n$  . מהי סיבוכיות זמן הריצה של האלגוריתם?

1.  $\Theta(n \log^2 n)$  3.  $\Theta(n \log^3 n)$

2.  $\Theta(n^2)$  4.  $\Theta(n \log^4 n)$

ג.

נתונה פונקציית זמן הריצה של אלגוריתם מסוים, הפועל על קלט שגודלו  $n$  :  
 $T(n) = 2T(n/2) + 7n$  . מהי סיבוכיות זמן הריצה של האלגוריתם?

1.  $\Theta(n)$  3.  $\Theta(n \log n)$

2.  $\Theta(n^2)$  4.  $\Theta(\log n)$

ד.

נתונה פונקציית זמן הריצה של אלגוריתם מסוים, הפועל על קלט שגודלו  $n$  :  
 $T(n) = 2T(n-1) - T(n-2) + 5n + 7$  . מהי סיבוכיות זמן הריצה של האלגוריתם?

1.  $\Theta(n^2)$  3.  $\Theta(n^2 5^n)$

2.  $\Theta(n^3)$  4.  $\Theta(n^4)$

ה.

נתונה פונקציית זמן הריצה של אלגוריתם מסוים, הפועל על קלט שגודלו  $n$  :  
 $T(n) = T(\sqrt{n}) + \log n$  . מהי סיבוכיות זמן הריצה של האלגוריתם?

1.  $\Theta(\sqrt{n} \cdot \log n)$

2.  $\Theta(\log n)$

3.  $\Theta(\sqrt{n})$

4.  $\Theta(n \log n)$

ו.

נתונה פונקציית זמן הריצה של אלגוריתם מסוים, הפועל על קלט שגודלו  $n$  :

$$T(n) = T\left(\frac{3n}{5}\right) + T\left(\frac{2n}{5}\right) + n$$

מהי סיבוכיות זמן הריצה של

האלגוריתם ?

1.  $\Theta(n \log n)$

2.  $\Theta(n)$

3.  $\Theta(n^2)$

4.  $\Theta(\log n)$



**בסעיפים ז' – י"א' התייחס לבעיה שלהלן:**

נתונות שתי קבוצות  $S_1$  ו- $S_2$ , המקיימות:

- I.  $S_1$  ו- $S_2$  זרות זו לזו.
  - II. כל קבוצה מכילה  $n$  מספרים שלמים חיוביים, כאשר  $n$  הוא כפולה שלמה של 3.
  - III. כל מספר  $c$  השייך לקבוצה  $S_1$ , קטן מכל מספר  $d$  השייך לקבוצה  $S_2$ .
  - IV. בכל קבוצה כל המספרים שונים זה מזה.
- נוסף על כך, נתון מספר שלם קבוע חיובי  $a$ .  
לפניך אלגוריתם יעיל הקובע אם קיימת קבוצה  $S_3$  שהיא תת-קבוצה של  $S_1$  (כלומר,  $S_3$  מוכלת ב- $S_1$ ), כאשר מתקיימים התנאים האלה:
- I.  $|S_3| \geq n/3$  (כמות המספרים שבקבוצה  $S_3$  תהיה לפחות  $n/3$ ).
  - II. לכל איבר  $x$  השייך לקבוצה  $S_2$  ולכל איבר  $y$  השייך לקבוצה  $S_3$  צריך להתקיים:  $x - y \geq a$ .

**הנח** שקיימת השגרה  $\text{partition}(S, n, z, S_1, S_2)$ , המבצעת חלוקה של הסדרה  $S$  בת  $n$  איברים לשתי סדרות,  $S_1$  ו- $S_2$ , שאינן ממוינות בהכרח. בסדרה האחת,  $S_1$ , יהיו האיברים הקטנים מ- $z$  או השווים ל- $z$  ובסדרה האחרת,  $S_2$ , יהיו האיברים גדולים מ- $z$ .

**האלגוריתם:**

```
temp = select (____(1)____, S1);          : צעד 1
partition (S1, n, ____ (2)____, S3, S4);    : צעד 2
max = select (n/3, ____ (3)____);           : צעד 3
min = ____ (4)____;                         : צעד 4
if ((min-max) ≥ a) הדפס תשובה חיובית;       : צעד 5
else הדפס תשובה שלילית;
```

באלגוריתם הנ"ל חסרים ארבעה ביטויים, המסומנים במספרים בין סוגריים עגולים. התשובה הנכונה עבור כל אחד מהביטויים החסרים מופיעה בסעיפים שלהלן:

**ז.**

מהי התשובה הנכונה בעבור ביטוי (1) לעיל?

1.  $n/3$
2. 1
3.  $2n/3$
4.  $n/2$

**ח.**

מהי התשובה הנכונה בעבור ביטוי (2) לעיל?

1.  $n/2$
2. temp
3.  $n/3$
4.  $2n/3$

**ט.**

מהי התשובה הנכונה בעבור ביטוי (3) לעיל?

1.  $S_2$
2.  $S_1 \cup S_2$
3.  $S_3$
4.  $S_4$

**י.**

מהי התשובה הנכונה בעבור ביטוי (4) לעיל?

1.  $\text{select}(2n/3, S_4)$
2.  $\text{select}(1, S_3)$
3.  $\text{select}(1, S_2)$
4.  $\text{select}(1, S_4)$



**י"א.**

מהי סיבוכיות זמן הריצה של האלגוריתם שתואר לעיל?

1.  $O(n \log n)$       3.  $O(n^2)$

2.  $O(n\sqrt{n})$       4.  $O(n)$

**י"ב.**

נתונה רשימה דו-מקושרת L, שאורכה אינה ידוע. אין לנו מצביעים לראשה ולזנבה של הרשימה, אבל יש לנו מצביע לאיבר מסוים S. מחפשים איבר X השייך לרשימה L שמיקומו ברשימה אינו ידוע.

מבצעים את שגרת החיפוש הבאה :

לכל  $K = 1, 2, \dots$  עוברים ברשימה K צעדים שמאלה וחזרה, אחר-כך K צעדים ימינה וחזרה, עד שמוצאים את X.

נסמן ב-n את המרחק של X מ-S.

מהי סיבוכיות זמן הריצה (במקרה הגרוע) של שגרה זו?

1.  $\Theta(n)$       3.  $\Theta(n^3)$

2.  $\Theta(n^2)$       4.  $\Theta(n^4)$

**י"ג.**

לפניך שגרה אחרת, שאף היא פותרת את הבעיה שהוצגה בסעיף הקודם (סעיף י"ב).

לכל  $K=2^i$ , כאשר  $i=1, 2, \dots$ , עוברים ברשימה K צעדים ימינה וחזרה, אחר-כך K צעדים

שמאלה וחזרה, עד ש  $K=n$ . n מבטא את המרחק של X מ-S.

מהי סיבוכיות זמן הריצה (במקרה הגרוע) של שגרה זו?

1.  $\Theta(\log n)$       3.  $\Theta(n)$

2.  $\Theta(n^2)$       4. מעריכית (אקספוננציאלית) כפונקציה של n.

**י"ד.**

נתון קטע הקוד הבא :

```
void func(int n)
{
    int i, j ;
    if (n<1) return ;
    for ( i= 1 ; i < n ; i*=2 ) j++ ;
    func(n - 1) ;
}
```

מהי סיבוכיות זמן הריצה של קטע הקוד הנתון כפונקציה של n ?

1.  $\Theta(n^2)$       3.  $\Theta(n)$

2.  $\Theta(\lg n)$       4.  $\Theta(n \lg n)$

**ט"ו.**

נתונה קבוצה P של n נקודות על הישר הממשי, ונקודה נוספת על הישר, q. המרחקים מ-q של כל

הנקודות ב-P שונים זה מזה. האלגוריתם היעיל ביותר שמוצא את  $\sqrt{n}$  הנקודות ב-P שהכי

קרובות ל-q רץ בזמן :



1.  $\Theta(n^2)$

2.  $\Omega(n \log n)$

3.  $O(n \log n)$

4.  $O(n)$

ט"ז. נתון קטע של קוד :

```
void func(int n)
{
    int i, j, temp=n, m=n ;
    while (temp!=0)
    {
        n=n+m;
        temp=temp/2;
    }
    for ( i= 0 ; i < n ; i++ )
        for ( j= 0 ; j<2019 ; j++ )
            printf("hi!");
}
```

מהי סיבוכיות זמן הריצה של קטע הקוד הנתון כפונקציה של n?

1.  $\Theta(n \log n)$       2.  $\Theta(\log n)$

3.  $\Theta(n^2)$       4.  $\Theta(n)$

י"ז. נתון קטע הקוד הבא :

```
void func(int n)
{
    int i, j ;
    for ( i= 1 ; i*i < n ; i++ )
        for ( j= 0 ; j*i<n ; j++ )
            printf("hi!");
}
```

מהי סיבוכיות זמן הריצה של קטע הקוד הנתון כפונקציה של n ?

1.  $\Theta(n^2)$       2.  $\Theta(n \lg n)$

3.  $\Theta(\lg n)$       4.  $\Theta(n)$

י"ח. נתון קטע הקוד הבא :

```
void func(int n)
{
    int i ;
    if (n<4) return ;
    for ( i= 0 ; i*i < n ; i++ ) { printf("hi!"); }
    for ( i= 0 ; i < 4 ; i++ ) { func(n/4) };
}
```

הנח כי כל המשתנים בקטע הם מטיפוס שלם.

מהי סיבוכיות זמן הריצה של קטע הקוד הנתון כפונקציה של n ?



- $\Theta(n^2)$  .1  
 $\Theta(n \lg n)$  .2  
 $\Theta(n)$  .3  
 $\Theta(\lg n)$  .4

י"ט. נתון קטע של קוד :

```
int func(int n , int x)
{
    if (x==0) return n ;
    int val;
    if ((x%2)!=0)
        val=func(n*2 , x/2);
    else
        val=func(n/2 , x/2);
    return val ;
}
```

```
void f10(int n)
{
    int m , i=  $\log_2 n$  ;
    m=func(4 , i );
    for ( i= 0 ; i < m ; i++ ) { printf("hi!") };
}
```

מהי סיבוכיות זמן הריצה של הפונקציה f10 כפונקציה של n?  
בחר את החסם ההדוק ביותר.

- $\Theta(n \log n)$  .1  
 $\Theta(\log \log n)$  .3  
 $\Theta(n)$  .2  
 $\Theta(\log n)$  .4

כ'. נתון קטע של קוד :

```
void func(int n)
{
    int i=1, j=0 ;
    while(j<n)
    {
        j=j+i ;
        i=i+i ;
    }
    for ( ; i*i < n ; i++ ) printf("hi!");
}
```

מהי סיבוכיות זמן הריצה של קטע הקוד הנתון כפונקציה של n ?



בחר את החסם ההדוק ביותר.

1.  $\Theta(n \log n)$

2.  $\Theta(n)$

3.  $\Theta(\log n)$

4.  $\Theta(\log \log n)$

**כ"א.** נתון קטע של קוד :

```
void f(int arr[], int n){
    int m=0;
    for(int i=1; i<n*n; ++i)
        for(int k=0; k<n*n; k+=i){
            m=(m>k)?m:k;
            printf("1");
        }
    free(malloc(m));
}
```

מהי סיבוכיות זמן הריצה של קטע הקוד הנתון כפונקציה של  $n$  ?  
בחר את החסם ההדוק ביותר.

1.  $\Theta(n^2 \log^2 n)$

2.  $\Theta(n^2 \log n)$

3.  $\Theta(n^2)$

4.  $\Theta(n^3)$

**כ"ב.** נתון קטע של קוד :

```
void func(int n)
{
    int j=n;
    while(j>1)
    {
        printf("hi!");
        j= log2 j;
    }
}
```

מהי סיבוכיות זמן הריצה של קטע הקוד הנתון כפונקציה של  $n$  ?  
בחר את החסם ההדוק ביותר.

1.  $\Theta(n \log n)$

2.  $\Theta(n)$

3.  $\Theta(\log^* n)$

4.  $\Theta(\log n)$



### שאלה 3

- א. האם הטענה הבאה נכונה תמיד? אם כן ענה ב-"נכונה" אחרת "לא נכונה".  
 if  $g(n) + f(n) \in \Omega(t(n))$  and  $g(n) - f(n) \in O(t(n))$  then  $f(n) \in \Omega(t(n))$   
 אם לדעתך הטענה נכונה אז הוכח אותה אחרת תנו דוגמא נגדית.  
 ב. הוכיחו או הפריכו:  $f(n) = \Theta(f(n/2))$   
 ג. הוכיחו או הפריכו:  $2^{O(\lg \lg n)} = O(\log n)$

### שאלה 4

- סדרו את הפונקציות לפי סדר אסימפטוטי  $O(\dots)$  (מן ה"קטנה" ל"גדולה"), אם שתי פונקציות בסידור מקיימות גם  $f_i = \Theta(f_k)$  אז ציינו זאת. הראה כי הסידור מקיים את הדרישה לכל זוג עוקב. הוכיחו את תשובתכם.  
 $n^{n+1}, (n+1)^n, 2^{2n}, 2^{\lg n}, 2^{\lg^2 n}, 3^n, 4^n, 4^{\sqrt{n}}, (4n)!, \lg((n^2)!), n^{10}, n^{\lg n}$

### שאלה 5

Find the Order of growth of the following functions:

- a)  $T(n) = T(an) + T(bn) + n \quad a > 0, b > 0, a + b < 1$   
 b)  $T(n) = T(\frac{n}{2} + \sqrt{n}) + \sqrt{6044}$   
 c)  $T(n) = T(\log n) + 1$   
 d)  $T(n) = \frac{1}{n} + \sum_{i=1}^{n-1} [T(i) - T(i-1)]$   
 $T(0) = 0, T(1) = 1$

**Prove your answers.**

### שאלה 6

Given a sorted array A of n distinct integers.

- a) Describe an  $O(1)$  time algorithm that determines whether or not there exists an element x, such that  $A[1] < x < A[n]$  and x is not in A.  
 b) Describe an  $O(\log n)$  time algorithm that finds such an x.  
 c) Describe an efficient algorithm that determines whether or not there exists an element i in A, such that  $A[i] = i$ . Analyze the running time of the algorithm.





## שאלה 7

תהי  $A$  מטריצת ריבועית בוליאנית (שערכיה הם 0 או 1) בגודל  $n$  על  $n$ . נגדיר את ערך ה-DNF של המטריצה כ-  $\bigvee_{i=1}^n \bigwedge_{j=1}^n A[i, j]$ , כלומר 1 אם ורק אם קיימת שורה במטריצה שכל איבריה הם 1-ים, ו-0 אחרת.

לדוגמא, עבור:

B=			A=		
0	1	1	0	1	1
1	1	1	1	1	0
0	1	0	0	1	0

נקבל  $DNF(A)=0$ ,  $DNF(B)=1$ .

הציעו מבנה נתונים התומך בפעולות הנתונות בסיבוכיות הנדרשת:

a.  $Init(n, A)$  – אתחל מטריצה  $A$  בגודל  $n$  על  $n$  שאיבריה כולם שווים ל-1.

סיבוכיות זמן  $O(n^2)$  במקרה הגרוע ביותר.

b.  $Flip(A, i, j)$  – אם  $A[i, j]=0$  הפוך אותו ל-1, ואם  $A[i, j]=1$  הפוך אותו ל-0.

סיבוכיות זמן  $O(1)$  במקרה הגרוע ביותר.

c.  $DNF(A)$  – החזר את ערך ה-DNF של  $A$ . סיבוכיות זמן  $O(1)$  במקרה הגרוע.

ממשו בשפת סי את מבני הנתונים ואת הפעולות.

## שאלה 8

לפניך הגדרה חדשה:

נתון מערך  $b = (b_0, b_1, \dots, b_i, b_{i+1}, b_{i+2}, \dots, b_{n-1})$ , שהוא מערך הממוין בסדר עולה

וללא איברים כפולים, כלומר:  $b_0 < b_1 < \dots < b_i < b_{i+1} < b_{i+2} < \dots < b_{n-1}$ .

מערך  $a$  ייקרא "מערך ממוין מוסט" כאשר מערך  $a$  מתקבל מן המערך הממוין  $b$  לאחר שכל איבר שהיה במערך  $b$  – הוסט  $i$  פעמים שמאלה ממקומו בצורה מעגלית, כאשר

$a = (b_i, b_{i+1}, \dots, b_{n-1}, b_0, b_1, b_2, \dots, b_{i-1})$ , כלומר:  $0 < i < n-1$ .

דוגמה:

עבור המערך הממוין הזה:

0	1	2	3	4
1	2	3	4	5



ה"מערך הממוין\_המוסט" המתאים בעבור  $i=2$ , יראה כך :

0	1	2	3	4
3	4	5	1	2

לפניך **מימוש יעיל** של פונקציה שכותרתה: `int findShifted(int arr[], int n, int x)`  
פונקציה זו מקבלת 2 פרמטרים :

**מערך ממוין\_מוסט** `arr` בגודל  $n$  המכיל מספרים השונים זה מזה ומספר נוסף  $x$ .  
אם הערך  $x$  קיים במערך `arr` אזי הפונקציה תחזיר את מיקומו במערך,  
אחרת – תחזיר את הערך  $(-1)$ .

בפונקציה זו חסרים **חמישה** ביטויים המסומנים במספרים בין סוגריים עגולים. בסעיף זה  
חמישה תתי סעיפים כאשר כל תת סעיף מייצג ביטוי חסר. עליך לענות על כל תתי הסעיפים. בכל  
תת סעיף נתונות ארבע תשובות, שרק אחת מהן נכונה. בכל תת סעיף בחר את התשובה הנכונה,  
והקף בעיגול את הספרה המייצגת אותה בדף התשובות שבנספח.

```
int findShifted(int arr[], int n, int x)
{
    int low = 0, high = n-1, mid = 0;
    int index;
    //arr[mid] < arr[mid-1] : שיתקיים ש:
    while (low <= high)
    {
        mid = (low+high) / 2;
        if(arr[n-1] <= arr[mid])
            low = mid+1;
        else if (arr[0] >= arr[mid])
            high = mid;
        if (_____(1)_____) break ;
    }
    index = _____(2)_____;
    if(_____(3)_____) return index;
    index = binarySearch(_____(4)____);
    return (index == -1)? -1 : _____(5)_____;
}
```

a. הביטוי החסר (1) הוא :

1. `arr[mid] == x`
2. `low == 0 && mid == high`
3. `low == mid`
4. `arr[mid] < arr[mid-1]`

b. הביטוי החסר (2) הוא :

1. `low`
2. `mid`
3. `binarySearch(arr+mid, n-mid+1, x)`
4. `binarySearch(arr, mid, x)`



c. הביטוי החסר (3) הוא :

1.  $low \leq high$
2.  $arr[mid] == x$
3.  $index \neq -1$
4.  $arr[index] == x$

d. הביטוי החסר (4) הוא :

1.  $arr, n, x$
2.  $arr+index, n-index, x$
3.  $arr+mid, n-mid, x$
4.  $arr, mid, x$

e. הביטוי החסר (5) הוא :

1.  $index$
2.  $index+mid$
3.  $(arr[mid] < x) ? index : mid+index$
4.  $(arr[0] > x) ? mid+index : index$

f. סיבוכיות זמן הריצה של הפונקציה הזאת היא :

1.  $\Theta(n^2)$
2.  $\Theta(\lg n)$
3.  $\Theta(n)$
4.  $\Theta(n \lg n)$

בהצלחה