

מבני נתונים ויעילות אלגוריתמים

הוראות לנבחנים

א. משך הבחינה: ארבע שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.

פרק ראשון: 40 נקודות

פרק שני: 60 נקודות

סך-הכול: 100 נקודות

ג. חומר עזר מותר בשימוש: כל חומר עזר כתוב בכתב-יד או מודפס על נייר.

ד. הוראות מיוחדות:

1. את התשובות לשאלות שבפרק הראשון יש לכתוב במחברת הבחינה.

את התשובות לשאלות שבפרק השני יש לסמן אך ורק על גבי דף התשובות שבנספח א'.

2. ענו על מספר השאלות הנדרש בשאלון. המעריך יקרא ויעריך את מספר התשובות הנדרש בלבד, לפי סדר כתיבתן, ולא יתייחס לתשובות נוספות.

3. בנספח ב' לשאלון זה מובא מילון מונחים בשפות עברית, ערבית, אנגלית ורוסית. תוכלו להיעזר בו בעת הצורך.

הוראות למשגיחים:

בתום הבחינה יש לוודא שהנבחנים הדביקו את מדבקת הנבחן שלהם
במקום המיועד לכך בדף התשובות שבנספח א', וצירפו אותו למחברת הבחינה.

כתבו במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונכם לכתוב כטייטה (ראשי פרקים, חישובים וכדומה).

כתבו "טייטה" בראש כל עמוד טייטה. כתיבת טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלולה לגרום לפסילת הבחינה!

בשאלון זה 27 עמודים ו-2 עמודי נספחים.

השאלות בשאלון זה מנוסחות בלשון רבים,

אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

פרק ראשון (40 נקודות)

שאלה 1 – שאלת חובה (40 נקודות)

בשאלה זו שלושה סעיפים (א'–ג'). עליכם לענות על כל הסעיפים.
בסעיפים שבהם יש שאלות רב-ברירה נתונות ארבע תשובות, אך רק אחת מהן נכונה. בכל שאלה העתיקו למחברת את התשובה הנכונה.
בסעיפים האחרים, פעלו על-פי ההנחיות.

חברת SKILLED DOGS עוסקת בהכשרת כלבים למשימות שונות, כגון: נחייה, שמירה, גישוש, תקיפה, ציד, גילוי סמים, גילוי חומרי נפץ ועוד.

החברה מחזיקה מתקן שבו נעשות כלל פעולותיה: מיון הכלבים המגיעים לתוכנית ההכשרה (ממקורות שונים); בחינת ההתאמה להכשרתם בכלל ולהתאמתם למשימות ספציפיות בפרט; ופעילויות ההכשרה השונות במסלולי הכשרה השונים, המותאמים לכל כלב ספציפי ולמשימותיו העתידיות. הפעילויות נעשות במסגרת קבוצות, שמספרן משתנה בהתאם לצורך ובהתאם לכלבים המשתתפים בהכשרה (הן מספר הקבוצות והן מספר הכלבים בכל קבוצה אינו קבוע). התוכנית כוללת מספר קבוצות להכשרה.

נסמן ב- K את מספר הקבוצות – שימו לב, מדובר במספר משתנה. כלב מועמד, שבעליו רשם אותו לתוכנית ההכשרה, עובר מבדק ראשוני של כישורים (כושר גופני, מבדק פסיכולוגי ומבדק אינטליגנציה). שקלול תוצאות המבדק משמש ציון לרמתו של הכלב, ועל פיו תיקבע מידת התאמתו להכשרה למשימות המתאימות לו במיוחד. בהמשך, במסגרת הקבוצה שאליה הוא שייך, הכלב נבחן וציונו מתעדכן בהתאם.

הנחות:

- i. כל כלב מזוהה על פי מספר זיהוי ושם.
- ii. לכלב תיק רפואי מסודר.
- iii. הכלב מלווה ברישום מדויק של תולדותיו ומוצאו.
- iv. הנהלת התוכנית היא המשבצת את הכלב באחת מקבוצות ההכשרה.
- v. הנהלת התוכנית רשאית להעביר כלב מקבוצה אחת לאחרת.
- vi. כלב לא יכול להיות משובץ ביותר מקבוצת הכשרה אחת.
- vii. הנהלת התוכנית רשאית לאחד שתי קבוצות הכשרה לקבוצה אחת.
- viii. בעליו של כלב רשאי בכל עת להפסיק את תוכנית ההכשרה של כלבו, ולעזוב את התוכנית.
- ix. הנהלת התוכנית רשאית לבטל קבוצת הכשרה כלשהי.
- x. בכל מחזור הכשרה של התוכנית מתקבלים N כלבים לכל היותר.

פעולות:

בחברת SKILLED DOGS מעוניינים לתכנן מערכת ממוחשבת שתאפשר את ניהול התוכנית. המערכת הממוחשבת תתמודד, בין היתר, בפעולות המפורטות בטבלה שלהלן:

הערה: הטבלה מכילה רק פעולות שייעשה בהן שימוש בהמשך.

1. הוספת כלב חדש למערכת addDog	הפעולה מוסיפה כלב חדש לקבוצה מסוימת במערכת הממוחשבת.
2. הסרת/העברת כלב מן המערכת removeDog	הפעולה מסירה / מעבירה כלב קיים מן המערכת הממוחשבת.

מבנה הנתונים:

1. מצביע לרשומה (struct) הראשית: RP

2. מבנה הרשומה הראשית (בטבלה להלן):

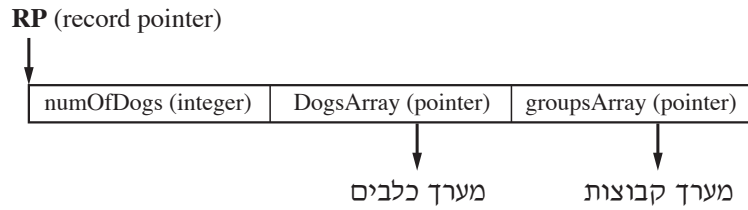
מזהה שדה	שם שדה	תיאור	הערות
A	numOfDogs	מס' הכלבים הרשומים ל"מחזור" ההכשרה	
B	DogsArray	מערך, בגודל N. בכל תא – פרטי כלב	זוהי טבלת גיבוב. הגישה באמצעות פונקציית HASH. ניתן להניח סיבוכיות של הפונקצייה $O(1)$
C	groupsArray	מערך, עבור הקבוצות. בכל תא – פרטי קבוצה מבין הקבוצות	תכולת התא: פרטי הקבוצה, ומצביע לשורש של עץ מאוזן ובו פרטים של ציוני הכלבים (עץ AVL). החיפוש באמצעות ציון (משמש כמפתח). הערות: • הציונים שונים • לכל קבוצה – עץ משלה

הגדרת קבוע (בשפת C):

```
#define N 100
```

הגדרת הרשומה (struct) הראשית (בשפת C):

```
typedef struct headType // טיפוס המבנה הראשי
{
    int numOfDogs; // מספר הכלבים בתוכנית (שדה A)
    DogsRec DogsArray[N]; // מערך כלבים (ראו שדה B)
    groupPtr groupsArray; // מערך של קבוצות הכשרה (ראו שדה C)
}
header, *headPtr;
```



מבנה הנתונים של מערך הכלבים: DogsArray

מבנה תא במערך (בשפת C):

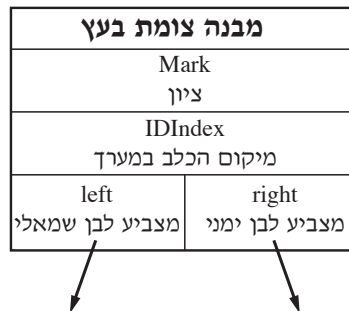
```
typedef struct dogType
{
    long dogID;           // מספר מזהה - מספר הזהות של הכלב
    char Name[20];        // שם הכלב
    int groupNo;          // מספר קבוצת הכשרה - האינדקס במערך הקבוצות
    MarkPtr sp;           // מצביע לצומת בעץ הציונים (*) של קבוצת הכלב
} DogsRec, *DogsPtr;
```

דוגמה למערך:

	dogID מספר מזהה של הכלב	Name שם הכלב	groupNo מספר הקבוצה שאליה שובץ	MarkPtr sp מצביע לצומת בעץ הציונים של קבוצת השיבוץ של הכלב
0				
1				
k	123456789	Bonbon	2	
m	124578902	Hercules	2	
t	761234990	Lilit	6	

(*) עץ ציונים: מבנה צומת (בשפת C):

```
typedef struct MarkType
{
    long Mark;                // ציון
    int IDIndex;              // מיקום הכלב במערך
    struct MarkType *left;    // מצביע לתת־העץ השמאלי
    struct MarkType *right;   // מצביע לתת־העץ הימני
}MarkRec, *MarkPtr;
```



מבנה הנתונים של קבוצה: groupsArray

מבנה תא במערך (בשפת C):

```
typedef struct groupType
{
    int DogsNum;           // מספר הכלבים בקבוצה
    float average;         // ממוצע (ציונים) בקבוצה
    long bestDog;          // ציון מרבי בקבוצה
    MarkPtr gp;            // מצביע לשורש העץ
}groupRec, *groupPtr;
```

דוגמה למערך:

	DogsNum מספר הכלבים בקבוצה	Average ממוצע הציונים בקבוצה	bestDog ציון מרבי בקבוצה	MarkPtr gp מצביע לצומת שורש העץ של ציוני הקבוצה
0				
1				
f	6	82.4	97	
h	5	70.2	88	
s	10	79.3	91	

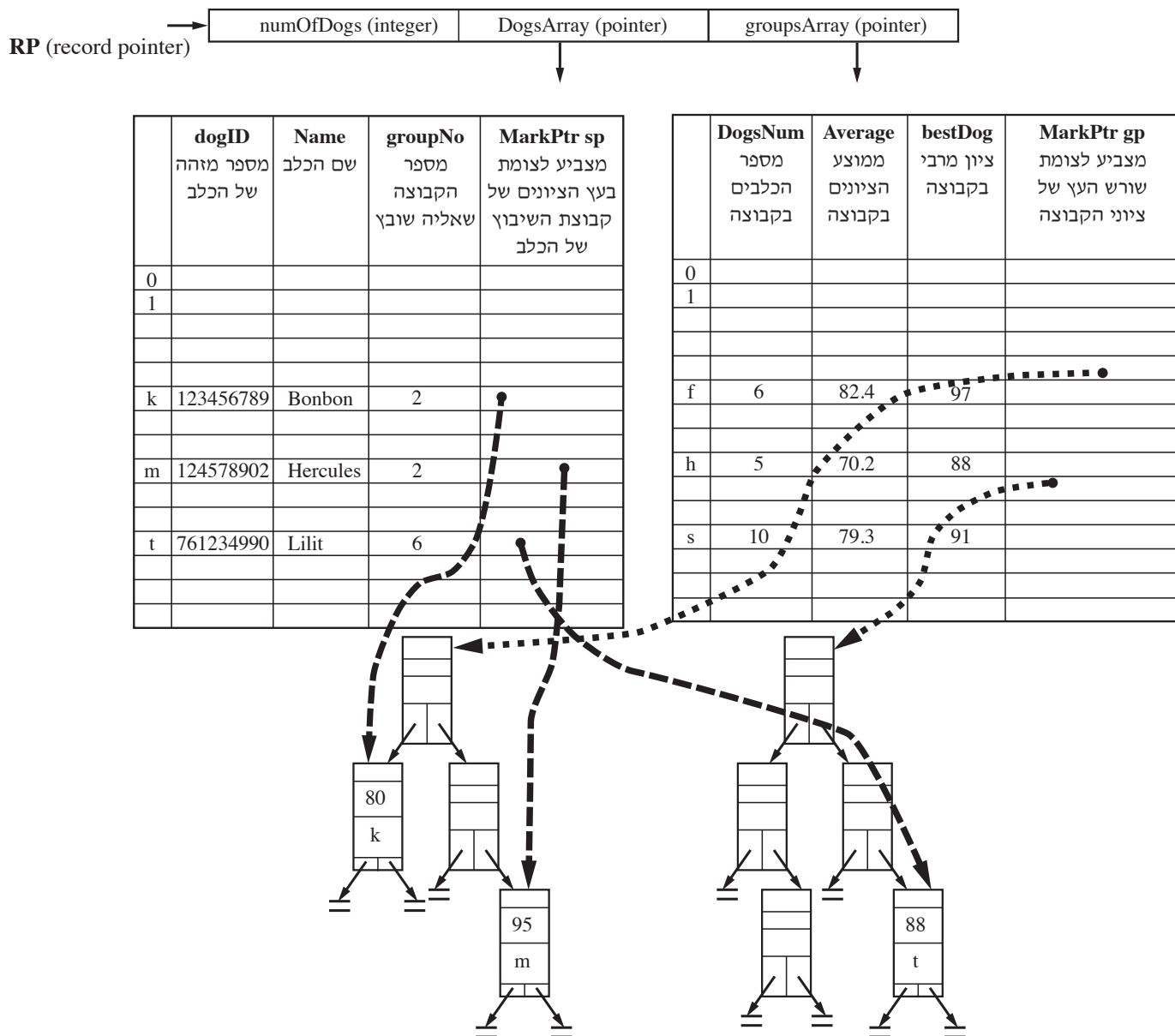
הגדרות כלליות:

```
typedef enum
{
    FAILURE, SUCCESS, INVALID_INPUT, ALLOCATION_ERROR, DUPLICATE_ID
} statusType;

typedef enum {FALSE, TRUE} Boolean;

typedef enum {GET, PUT} modType;
```

תיאור הקשרים והמצביעים:



נתונה ספריית פונקציות המכילה, בין היתר, את הפונקציות שלהלן:

1	<code>int hash(long num, modType mode)</code> פונקצייה זו מקבלת שני פרמטרים: • <code>num</code> מספר שלם • <code>mode</code> שיכול לקבל את הערכים: • PUT (במקרה זה הפונקצייה מחזירה מיקום בטבלת HASH, לשם תבוצע הכנסת הערך <code>num</code>) • GET (במקרה זה הפונקצייה מחפשת בטבלת HASH את הערך <code>num</code> – אם נמצא, מוחזר מיקומו בטבלה. אחרת, מוחזר ערך שלילי -1) <u>הערות:</u> • במקרה של הכנסת ערך, הניחו כי תמיד יימצא מקום בטבלת HASH • סיבוכיות זמן הריצה של הפונקצייה: $O(1)$
2	<code>statusType insertTree(MarkPtr *pl, MarkPtr t)</code> הפונקצייה מקבלת שני פרמטרים: • <code>pl</code> – כתובת של מצביע לשורשו של עץ ציונים (עץ קבוצה) כלשהו • <code>t</code> – מצביע לצומת בעץ המייצג "כלב" (הניחו בשאלה זו קיומו של עץ) הפונקצייה מוסיפה לעץ הציונים את הצומת שעליו מצביע <code>t</code>, ומחזירה את הערך <code>SUCCESS</code> <u>הערות:</u> 1. הכלב לא קיים בעץ הציונים הזה 2. לאחר הוספת הצומת, עץ ציונים יישאר עץ AVL 3. סיבוכיות זמן הריצה של הפונקצייה: $O(\log n)$
3	<code>MarkPtr deleteFromMarksTree(MarkPtr *t, MarkPtr p)</code> הפונקצייה מקבלת שני פרמטרים: • <code>t</code> – כתובת של מצביע לשורש של עץ ציונים כלשהו • <code>p</code> – מצביע לצומת בעץ הציונים הזה הפונקצייה מבצעת: 1. יצירת צומת חדש שעליו מצביע <code>q</code> (מצביע עזר). צומת זה יכיל את כל המידע המשוכן בצומת שעליו מצביע <code>p</code> 2. הוצאת הצומת שעליו מצביע <code>p</code> מעץ הציונים 3. מחזירה מצביע לצומת שעליו מצביע <code>q</code> <u>הערה:</u> לאחר הוצאת הכלב, עץ הציונים יישאר עץ AVL

הניחו שהפונקציות האלה כתובות וניתן להשתמש בהן בכל הסעיפים הבאים בלי לכתוב אותן מחדש. כמו כן, בעבור כל סעיף תוכלו להשתמש בכל פונקצייה שמומשה בסעיפים שלפניו.

להלן הגדרות של המשתנים הגלובליים בתוכנית:

```
header head;  
  
headPtr RP = &head;  
  
int numOfGroup;          // מספר הקבוצות שקיימות בזמן נתון במערכת הממוחשבת
```

(14 נק') א. להלן פונקציית שכתרתה:

```
statusType addDog(long ID, char Name[], int group, long Mark)
```

הפונקצייה מקבלת את הפרמטרים האלה:

ID – מספר הזהות של הכלב

Name – שם הכלב

group – מספרה של הקבוצה שבה משובץ הכלב

Mark – הציון של הכלב

הפונקצייה מוסיפה כלב חדש, המזוהה ב-ID, למערכת הממוחשבת.
נוסף על כך, הפונקצייה מעדכנת את מספר הכלבים הרשומים במערכת.
הפונקצייה מחזירה ערך מטיפוס statusType כמפורט בטבלה שלהלן:

הערך המוחזר מטיפוס status Type	תיאור (משמעות ההודעה)
ALLOCATION_ERROR	קיימת בעיה בהקצאת זיכרון.
DUPLICATE_ID	קיים כלב בעל מזהה ID בטבלת הערבוד.
INVALID_INPUT	הקבוצה שמספרה group לא קיימת.
SUCCESS	הכלב המזוהה ב-ID נוסף בהצלחה למערכת.

1. בפונקציית חסרים **שישה** ביטויים, המסומנים במספרים בין סוגריים עגולים. כתבו במחברת את מספרי הביטויים החסרים (1)–(6), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
statusType addDog(long ID, char Name[], int group, long Mark)
{
    statusType status = SUCCESS;

    MarkPtr p;

    int sindx;

    int DogsNum;

    float average;

    sindx = hash(ID,GET);
    if(_____ (1) _____) return DUPLICATE_ID;
    sindx = _____ (2) _____;
    if(group < 0 || group >= numOfGroup) return INVALID_INPUT;
    RP->DogsArray[sindx].dogID = ID;
    strcpy(RP->DogsArray[sindx].Name,Name);
    RP->DogsArray[sindx].groupNo = group;
    p = malloc(sizeof(MarkRec));
    if(p == NULL) return ALLOCATION_ERROR;
    p->Mark = Mark;
    p->IDIndex = sindx;
    _____ (3) _____ = p;
    DogsNum = RP->groupsArray[group].DogsNum;
    average = RP->groupsArray[group].average;
    average = (average*DogsNum+ Mark) / (DogsNum +1);
    RP->groupsArray[group].average = average;
    _____ (4) _____;
    if(Mark> _____ (5) _____) _____ (5) _____ = Mark;
    status = insertTree(_____ (6) _____);
    RP->numOfDogs++;
    return status;
}
```

2. מהי סיבוכיות זמן הריצה של הפונקצייה? העתיקו למחברתכם את התשובה הנכונה.
הבהרה: k מציין את מספר הכלבים בקבוצה שבה משובץ הכלב, n מציין את מספר הכלבים הרשומים במערכת הממוחשבת.

1. $O(\log(n + k))$

2. $O(1)$

3. $O(\log k)$

4. $O(k)$

18 נק') **ב.** להלן פונקצייה שכותרתה: `statusType removeDog(long dogID)`
הפונקצייה מקבלת את הפרמטר `dogID` (מספר הזהות של כלב מסוים), ומסירה מן המערכת את הכלב שמספר הזיהוי שלו התקבל. לאחר מכן, ערכו של `dogID` שהתקבל ישתנה ל-0 במערך `DogsArray`.
נוסף על כך, הפונקצייה מעדכנת את מספר הכלבים הרשומים במערכת זו, וכן את יתר הפרטים בעבור הקבוצה שבה היה משובץ הכלב שהוסר מן המערכת.
לתשומת לבכם:
לאחר הסרת הכלב, הקבוצה שממנה הוסר יכולה להישאר במערכת הממוחשבת כקבוצה ריקה.

הפונקצייה מחזירה ערך מטיפוס `statusType` כמפורט בטבלה שלהלן:

הערך המוחזר מטיפוס <code>status Type</code>	תיאור (משמעות ההודעה)
INVALID_INPUT	הכלב שמזוהה באמצעות <code>dogID</code> לא קיים במערכת.
SUCCESS	הכלב המזוהה ב-ID הוסר בהצלחה מהמערכת.

1. בפונקצייה חסרים **שמונה** ביטויים, המסומנים במספרים בין סוגריים עגולים. כתבו במחברת את הביטויים החסרים (1)-(8), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
statusType removeDog(long dogID)
{
    statusType status = SUCCESS;
    MarkPtr t,s,q;
    int indx,group;
    DogsRec scr;
    groupRec gpr;
    long Mark;
    indx = hash(dogID,GET);
    if(indx == -1) return INVALID_INPUT;
    RP->DogsArray[indx].dogID = 0;
    group = _____(1)_____;
```

```
scr = RP->DogsArray[indx];
gpr = RP->groupsArray[group];
Mark = scr._____(2)_____;
if(gpr.DogsNum == 1) gpr.average = 0;
else gpr.average = (gpr.average * gpr.DogsNum - Mark) / (gpr.DogsNum - 1);
_____(3)_____ = gpr.average;
RP->groupsArray[group].DogsNum--;
s = _____(4)_____;
t = _____(5)_____;
q = deleteFromMarksTree(&t,s);
RP->groupsArray[group].gp = t;
if(!t) RP->groupsArray[group].bestDog=0;
else
{
    if(q-> Mark == gpr.bestDog)
    {
        while(_____(6)_____)
            t = _____(6)_____;
        RP->groupsArray[group].bestDog = _____(7)_____;
    }
}
scr.dogID = 0;
_____(8)_____;
return status;
}
```

2. מהי סיבוכיות זמן הריצה של הפונקצייה? העתיקו למחרתכם את התשובה הנכונה.
הבהרה: k מציין את מספר הכלבים בקבוצה שבה משובץ הכלב, n מציין את מספר הכלבים הרשומים במערכת הממוחשבת.

1. $O(\log k)$

2. $O(1)$

3. $O(\log(n + k))$

4. $O(k)$

8 נק') ג. להלן פונקצייה (רקורסיבית) שכותרתה: void **printTree**(MarkPtr t)

הפונקצייה מקבלת את הפרמטר t, אשר מצביע לשורשו של עץ ציונים השייך לקבוצה כלשהי, ומדפיסה בעבור כל כלב בקבוצה זו את שמו ואת ציונו.

שמות הכלבים וציוניהם יודפסו באופן ממוין, בסדר עולה על פי הציון.

1. בפונקצייה חסרים שלושה ביטויים, המסומנים במספרים בין סוגריים עגולים. כתבו במחברת את הביטויים החסרים (1)-(3), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
void printTree(MarkPtr t)
{
    DogsRec sr;
    if (t!= NULL)
    {
        _____ (1) _____;
        sr = _____ (2) _____;
        printf ("\n%s\t%ld\n", sr.Name, sr.sp->Mark) ;
        _____ (3) _____;
    }
}
```

2. מהי סיבוכיות זמן הריצה של הפונקצייה? העתיקו למחברתכם את התשובה הנכונה.
הבהרה: k מציין את מספר הכלבים בקבוצה שבה משובץ הכלב, n מציין את מספר הכלבים הרשומים במערכת הממוחשבת.

1. $O(\log(n + k))$

2. $O(1)$

3. $O(\log k)$

4. $O(k)$

פרק שני (60 נקודות)

בפרק זה כל השאלות הן שאלות רב-ברירה.

ענו על שלוש מבין השאלות 2-5 (לכל שאלה – 20 נקודות).

סמנו את התשובות לשאלות שבפרק זה אך ורק על-גבי דף התשובות שבנספח א'.

שאלה 2

ענו על כל הסעיפים א'-י' (לכל סעיף – 2 נקודות).

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

$$T(n) = T(\sqrt{n}) + 5$$

מהי סיבוכיות זמן הריצה של האלגוריתם?

1. $O(\log n)$

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

3. $O(\sqrt{n} \log n)$

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

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

$$T(n) = 3T\left(\frac{n}{4}\right) + 5n$$

מהי סיבוכיות זמן הריצה של האלגוריתם?

1. $O(n)$

2. $O(n^{\log_4 3})$

3. $O(n \log n)$

4. $O(n^{\log_3 4})$

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

$$T(n) = 2T(n-1) - T(n-2) + 5n + 1$$

מהי סיבוכיות זמן הריצה של האלגוריתם?

1. $O(n)$

2. $O(n5^n)$

3. $O(n^3)$

4. $O(n^2 5^n)$

ד. נתון קטע הקוד הזה:

```
for (i=1; i<n; i++)  
{  
    while (i>1) i/=5;  
    while (i<n) i+=2;  
}
```

נתון כי כל המשתנים שבקטע הקוד הם מטיפוס שלם.

מהי סיבוכיות זמן הריצה של קטע קוד זה, כפונקצייה של n ?

1. $O(n^2)$

2. $O(\log n)$

3. $O(n)$

4. $O(n \log n)$

ה.

נתונה רשימה דו־כיוונית L ומצביע לאיבר מסוים S ברשימה.

מחפשים איבר X ברשימה, ומבצעים את החיפוש הבא: לכל $K = 1, 2, 3, \dots$ עוברים ברשימה K צעדים שמאלה וחזרה, אחר כך K צעדים ימינה וחזרה, עד שמוצאים את X . כל צעד מעביר אותנו לאיבר הבא.

יהי n המרחק (בצעדים) בין X ל־ S .

מהי סיבוכיות זמן הריצה של אלגוריתם חיפוש זה כפונקצייה של n ?

1. $O(n)$

2. $O(\log n)$

3. $O(n \log n)$

4. $O(n^2)$

ו.

נתונה רשימה דו־כיוונית L ומצביע לאיבר מסוים S ברשימה.

מחפשים איבר X ברשימה, ומבצעים את החיפוש הבא: לכל $K = 1, 2, 2^2, 2^3, \dots$ עוברים ברשימה K צעדים שמאלה וחזרה, אחר כך K צעדים ימינה וחזרה, עד שמוצאים את X . כל צעד מעביר אותנו לאיבר הבא.

יהי n המרחק (בצעדים) בין X ל־ S .

מהי סיבוכיות זמן הריצה של אלגוריתם חיפוש זה כפונקצייה של n ?

1. $O(n)$

2. $O(\log n)$

3. $O(n \log n)$

4. $O(n^2)$

ז. נתונה פונקציית זמן הריצה של אלגוריתם: $T(1) = 1, T(n) = n + \sum_{i=1}^{n-1} T(i)$

מהי סיבוכיות זמן הריצה של האלגוריתם?

1. $O(n^2)$

2. $O(n^3)$

3. $O(n \log n)$

4. $O(2^n)$

ח. יהי $f(n)$ מספר סדרות בגודל n שאפשר ליצור בעזרת א"ב $\{0,1,2\}$ ללא אחדים צמודים.

מהי הנוסחה הרקורסיבית בעבור $f(n)$?

1. $f(n) = f(n-1) + f(n-2)$

2. $f(n) = 2f(n-1) + f(n-2)$

3. $f(n) = f(n-1) + 2f(n-2)$

4. $f(n) = 2f(n-1) + 2f(n-2)$

סעיפים ט' ו-י' מתייחסים לקטע הקוד הבא:

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

ט. מהי הנוסחה הרקורסיבית עבור זמן הריצה של פונקציית f ?

1. $T(n) = T(n-1) + \frac{n}{2}$

2. $T(n) = T(n-1) + \log n$

3. $T(n) = \log n T(n-1) + 1$

4. $T(n) = T(n-1) + n$

י. מהי סיבוכיות זמן הריצה של קטע קוד זה כפונקצייה של n ?

1. $O(n)$

2. $O(\log n)$

3. $O(n \log n)$

4. $O(n^2)$

שאלה 3

ענו על כל הסעיפים א'-י' (לכל סעיף - 2 נק').

סעיפים א'-ג' מתייחסים לבעיה שלהלן:

לפניכם אלגוריתם המקבל מערך S של n אותיות קטנות באנגלית, ומחזיר את מספר ה"מילים" (רצף תווים) המתחילות באות a ומסתיימות באות b .

לדוגמה: עבור המערך $S=\{a,a,b,a,x,b,c,d\}$ יהיה הפלט 5 ($aab, ab, axb, aabaxb, abaxb$).

Number AB-words(S, n)

```
1. i = 0;
2. counter_1 = 0;
3. counter_2 = 0;
4. while (i < n)
    4.1 if (S[i] == 'a')
        _____ (1) _____;
        else
            if (S[i] == 'b')
                _____ (2) _____;
    4.2 i=i+1;
5. return _____ (3) _____;
```

א. הביטוי החסר (1) הוא:

- 1. counter_1=1
- 2. counter_2=counter_1+1
- 3. counter_1=counter_1+1
- 4. counter_2=counter_2+ counter_1

ב. הביטוי החסר (2) הוא:

- 1. counter_2=1
- 2. counter_2= counter_2+1
- 3. counter_1=counter_1+1
- 4. counter_2=counter_2+ counter_1

ג. הביטוי החסר (3) הוא:

1. counter_1
2. counter_2
3. counter_1 + counter_2
4. counter_2 - counter_1

ד. נתון הביטוי $a/b + c*d$ בצורת postfix. מאיזה מהביטויים שלהלן, בצורת prefix, מתקבל הביטוי?

1. $a/b - c*b*d$
2. $a - c*b*d/t$
3. $+ - a/c/*b*d$
4. $+ / t - * d b c a$

ה. בטקסט מעל האלף-בית $\Sigma = \{a, b, c, d, e\}$ האות a מופיעה שמונה (8) פעמים, האות b פעמיים (2), האות c עשר (10) פעמים, האות d ארבע (4) פעמים, והאות e - פעמים x .

קוד הופמן של האותיות הוא:

$a - 10, b - 000, c - 11, d - 001, e - 01$

מהו הערך האפשרי של x ?

1. 3
2. 7
3. 10
4. 14

ו. מה ניתן לומר על עץ בינארי שיש לו אותן פריסות ב-"postorder" וב-"inorder"?

1. תת-העץ הימני של כל צומת בעץ בינארי זה הוא עץ ריק.
2. תת-העץ השמאלי של כל צומת בעץ בינארי זה הוא עץ ריק.
3. בעץ בינארי זה יש צומת אחד לכל היותר.
4. עץ בינארי זה הוא בהכרח עץ ריק.

סעיפים ז'-י' מתייחסים לבעיה שלהלן:

צומת בעץ בינארי מוגדר כך:

```
typedef struct treenode{  
  
    int data;  
  
    struct treenode *left;  
  
    struct treenode *right;  
  
} tree_node, *tree;
```

נתונה ספרייה הכוללת הגדרות ופונקציות לטיפול במחסניות.

יהי PStack_int מצביע למחסנית המאחסנת מספרים שלמים,

ויהי PStack_ptr מצביע למחסנית המאחסנת מצביעים לעצים בינאריים.

לפניכם קוד של פונקצייה, המקבלת מצביע על עץ בינארי ומחזירה את גובהו של העץ.

גובה של עץ הוא רמה מרבית של צומת כלשהו בעץ, ורמתו של שורש היא אפס.

הניחו כי העץ לא ריק.

```
int gova2(tree p)  
{  
    tree E;    int L,T;  
    PStack_ptr S1; PStack_int S2;  
    Init(S1); Init(S2);  
    Push(S1,p); Push(S2,0);  
    T=0;  
    while(! IsEmpty(S1))  
    {  
        E=Pop(S1); L=Pop(S2);  
        if (T < L ) _____ (1) _____;  
        if (E->right)  
        {  
            Push(S1, E->right);  
            Push(S2, _____ (2) _____);  
        }  
        if (E->left)
```

```

    {
        Push(S1, E->left);
        _____ (3) _____ ;
    }
}
_____ (4) _____ ;
}

```

ז. הביטוי החסר (1) הוא:

1. $T=L$

2. $T=T+1$

3. $T=T+L$

4. $L=L-1$

ח. הביטוי החסר (2) הוא:

1. T

2. L

3. $L+1$

4. $T+1$

ט. הביטוי החסר (3) הוא:

1. $Push(S2, L)$

2. $Push(S2, L+1)$

3. $Push(S2, T)$

4. $Push(S2, T+1)$

י. הביטוי החסר (4) הוא:

1. $return Pop(S2)$

2. $return T+L$

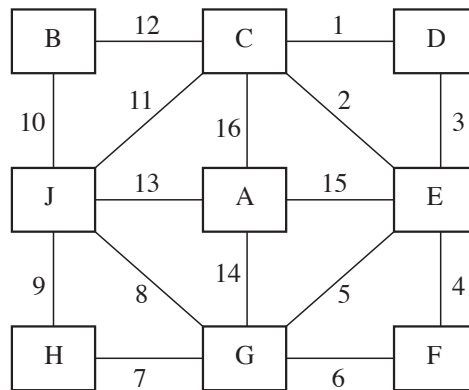
3. $return L$

4. $return T$

שאלה 4

ענו על כל הסעיפים א'–י' (לכל סעיף – 2 נק').

א. להלן גרף.



מהו סדר המשקלים של צלעות העץ ש-PRIM הכניס אותם לעץ פורש מינימלי החל מקודקוד A? יש לקרוא משמאל לימין.

1. 13,8,7,5,4,2,1,12

2. 13,8,5,4,2,1,7,10

3. 13,8,5,4,2,1,12,7

4. 13,8,5,2,1,4,7,10

סעיפים ב'–ד' מתייחסים לבעיה שלהלן:

נתון מערך A של n מספרים טבעיים שונים, ומספר טבעי k הקטן מ- n אך גדול מ- $n/2$.
נסמן $\text{Sum}(A)$ – סכום איברים במערך A .

לפניכם אלגוריתם בסיבוכיות $O(n)$, המחלק את המערך A לשני מערכים, B ו- C , בגודל k ו- $n-k$ (בהתאמה), כך שההפרש בין שני סכומי האיברים במערכים B ו- C יהיה מרבי:

$$|\text{Sum}(B) - \text{Sum}(C)| \rightarrow \max$$

$\text{Max_Partition}(A, n, k)$

```
1. j=1, t=1;
2. M= _____ (1)
3. for(i=1; i<=n; i++)
    if(A[i] > M)
        _____ (2)
    else
        _____ (3)
4. return B, C
```

ב. הביטוי החסר (1) הוא:

1. $\text{Select}(A, (n+1)/2)$
2. $\text{Select}(A, k)$
3. $\text{Select}(A, n-k)$
4. $\text{Select}(A, k - (n/2))$

ג. הביטוי החסר (2) הוא:

1. $B[j] = A[i]; j = j+1;$
2. $C[j] = A[i]; j = j+1;$
3. $B[j] = A[i]; i = i+1;$
4. $C[t] = A[i]; t = t+1;$

ד. הביטוי החסר (3) הוא:

1. $B[t] = A[i]; t = t+1;$
2. $B[j] = A[i]; j = j+1;$
3. $C[j] = A[i]; j = j+1;$
4. $C[t] = A[i]; t = t+1;$

סעיפים ה'-ו' מתייחסים לבעיה שלהלן:

נתון עץ AVL לאחסון מספרים שלמים עם שדה נוסף לאחסון חזרות: לצד המספר השלם, המשמש כמפתח בעץ, נשמר מספר החזרות שלו בקלט.

ה. מה תהיה סיבוכיות ההכנסה לעץ במקרה **הגרוע ביותר** (W.C.) עבור קלט המכיל n מספרים שלמים, כאשר מתוכם יש רק $\sqrt{n}$ מספרים שונים?

1. $O(\sqrt{n} \log n)$

2. $O(n)$

3. $O(n \log n)$

4. $O(\sqrt{n})$

ו. מה תהיה סיבוכיות ההכנסה לעץ במקרה **הטוב ביותר** (B.C.) עבור קלט המכיל n מספרים שלמים, כאשר מתוכם יש רק $\sqrt{n}$ מספרים שונים?

1. $O(\sqrt{n} \log n)$

2. $O(n)$

3. $O(n \log n)$

4. $O(\sqrt{n})$

ז. לפניכם ארבע סדרות המייצגות הדפסה במעבר postorder על עץ בינארי. איזו סדרה מביניהן יכולה להיות מייצגת של עץ חיפוש בינארי, אך **שאינו** AVL?

1. 201, 154, 309, 100, 402, 56, 505

2. 201, 154, 505, 309, 100, 56, 402

3. 56, 154, 100, 309, 505, 402, 201

4. 56, 309, 154, 100, 505, 402, 201

ח. יהי עץ חיפוש בינארי שהוכנסו אליו המפתחות הבאים (יש לקרוא משמאל לימין):

50, 4, 48, 6, 87, 29, 81, 56, 11

מהי הטענה שאינה נכונה?

1. בתת-העץ הימני יש שלושה צמתים.
2. לעץ זה יש שני עלים.
3. הבן הימני של הצומת בעל המפתח 4 הוא 48.
4. הבן השמאלי של הצומת בעל המפתח 48 הוא 29.

ט. יהי N_h מספר מזערי של צמתים בעץ AVL בעל גובה h .

הניחו כי תת-העץ השמאלי בגובה $h-1$,

תת-העץ הימני בגובה $h-2$, וכל תתי-העצים הם עצי AVL.

שימו לב: $N_0 = 0, N_1 = 1$.

מהי נוסחת הנסיגה עבור N_h ?

1. $N_{h-1} + N_{h-2}$
2. $N_{h-1} + N_{h-2} + 1$
3. $N_{h-1} + 1$
4. $N_{h-1} + h - 1$

י. יהי $G=(V,E)$ גרף מכוון המיוצג על ידי המטריצה הבאה:

0	0	1	0	0
0	0	0	1	0
0	1	0	0	1
0	0	0	0	0
1	0	0	0	0

מהו המספר המזערי של קשתות שיש להוסיף לגרף הנתון, כדי שיהפוך לגרף קשיר חזק (המכיל רכיב קשירות חזק אחד)?

1. קשת אחת
2. שתי קשתות
3. שלוש קשתות
4. ארבע קשתות

שאלה 5

ענו על כל הסעיפים א'–ח' (לכל סעיף – 2.5 נק').

א. מהי הטענה שאינה נכונה?

1. יהי C מעגל בגרף מכוון. מפעילים אלגוריתם DFS על הגרף. אם v קודקוד ראשון מבין הקודקודים של מעגל C שגילה האלגוריתם DFS, אזי כל הקודקודים של מעגל C נמצאים בתת-העץ של עץ DFS שקודקוד v הוא שורשו.
2. נתון גרף שלם לא מכוון K_n עם פונקציית משקל $w: E \rightarrow R^+$. אם יש לגרף הזה עץ"מ יחיד, אזי לכל הקשתות שלו יש בהכרח משקל שונה.
3. יהי $G=(V,E)$ גרף קשיר לא מכוון. יהי $d[v]$ זמן גילוי הצומת v הנקבע על ידי הרצת אלגוריתם DFS. אם נכוון כל צלע (u,v) בגרף, מהקודקוד שגילינו קודם לקודקוד שגילינו לאחר מכן, נקבל גרף מכוון חסר מעגלים מכוונים (גמ"ל).
4. יהי $G=(V,E)$ גרף לא מכוון. אם בגרף זה יש בדיוק שני קודקודים v ו- u עם דרגה אי-זוגית, אז הם שייכים לאותו רכיב קשירות.

סעיפים ב'–ו' מתייחסים לבעיה שלהלן:

נתון גרף קשיר ולא מכוון $G=(V,E)$ וצומת $s \in V$.
יהי $d[v]$ המרחק המזערי מצומת s לצומת v בגרף G ,
ויהי $p[v]$ מספר המסלולים הקצרים ביותר מצומת s לצומת v בגרף G . $Adj[u]$ מסמן קבוצת שכנים של u .
לפניכם אלגוריתם המוצא לכל צומת v בגרף G את $d[v]$ ואת $p[v]$.

```
BFS_update(G, s)
FOR each  $x \in V - [s]$  {
    color[x] = WHITE;
    d[x] =  $\infty$  ; p[x] = 0; }
color[s] = GRAY; d[s] = 0; p[s] = 1;
Q =  $\emptyset$ ;
ENQUEUE(Q, s);
```

```

WHILE not IsEmpty(Q) {
    u = DEQUEUE(Q);
    FOR each  $v \in Adj[u]$  {
        IF color[v] == _____ (1) {
            color[v] = GRAY;
            d[v] = d[u]+1;
            _____ (2);
            ENQUEUE(Q, v);
        }
        ELSE
            IF _____ (3)
                _____ (4);
    } //end FOR
    color[u] = BLACK;
} //end WHILE
RETURN d[ ], p[ ];

```

ב. הביטוי החסר (1) הוא:

- 1. BLACK
- 2. GRAY
- 3. WHITE
- 4. color[u]

ג. הביטוי החסר (2) הוא:

- 1. $p[v] = p[u]$
- 2. $p[v] = p[u] + 1$
- 3. $p[v] = p[v] + 1$
- 4. $p[v] = 1$

ד. הביטוי החסר (3) הוא:

- 1. $color[v] == GRAY$
- 2. $color[v] == BLACK$
- 3. $d[v] == d[u]$
- 4. $d[v] == d[u] + 1$

ה. הביטוי החסר (4) הוא:

1. $p[v] = p[u]$

2. $p[v] = p[u] + 1$

3. $p[v] = p[v] + p[u]$

4. $p[v] = 1$

ו. מהי סיבוכיות האלגוריתם?

1. $O(|V|^2)$

2. $O(|V| * |E|)$

3. $O(|V|)$

4. $O(|V| + |E|)$

ז. נתונה ערמת מקסימום, בגודל n , של מפתחות טבעיים. ידוע ש- $n = 2^k$.

מפתח השורש הוא n , והמפתח של כל צומת קטן פי 2 ממפתח האב שלו. להלן שתי טענות:

A: סכום המפתחות של העלים הוא $O(n)$.

B: סכום כל המפתחות הוא $O(n \log n)$.

איזה מבין ההיגדים שלהלן הוא ההיגד הנכון?

1. שתי הטענות נכונות.

2. שתי הטענות לא נכונות.

3. טענה A נכונה, וטענה B לא נכונה.

4. טענה A לא נכונה, וטענה B נכונה.

ח. ממיינים סדרה, בגודל n , של מספרים טבעיים. ידוע שסדרת המספרים כוללת $(n-1)$ אחדים, והופעה אחת של מספר 2. איזו מבין הטענות הבאות אינה נכונה?

1. אם השתמשנו באלגוריתם מיון-מיזוג (merge-sort), אזי זמן הריצה הוא $O(n \log n)$ בהכרח.

2. אם השתמשנו באלגוריתם מיון-מהיר (quick-sort), אזי זמן הריצה הוא $O(n \log n)$ בהכרח.

3. אם השתמשנו באלגוריתם מיון-ערמה (heap-sort), אזי זמן הריצה הוא $O(n \log n)$.

4. אם השתמשנו באלגוריתם מיון-מנייה (counting-sort), אזי זמן הריצה הוא $O(n)$.

בהצלחה!

יש לענות על שלוש מבין השאלות 2-5 (לכל שאלה - 20 נקודות).

הקיפו בעיגול את הספרה המייצגת את התשובה הנכונה לכל סעיף.

הדביקו את מדבקת הנבחן במקום המיועד לכך, והדקו את הדף הזה למחברת הבחינה.

שאלה 2					שאלה 3				
סעיף א	1	2	3	4	סעיף א	1	2	3	4
סעיף ב	1	2	3	4	סעיף ב	1	2	3	4
סעיף ג	1	2	3	4	סעיף ג	1	2	3	4
סעיף ד	1	2	3	4	סעיף ד	1	2	3	4
סעיף ה	1	2	3	4	סעיף ה	1	2	3	4
סעיף ו	1	2	3	4	סעיף ו	1	2	3	4
סעיף ז	1	2	3	4	סעיף ז	1	2	3	4
סעיף ח	1	2	3	4	סעיף ח	1	2	3	4
סעיף ט	1	2	3	4	סעיף ט	1	2	3	4
סעיף י	1	2	3	4	סעיף י	1	2	3	4

שאלה 4					שאלה 5				
סעיף א	1	2	3	4	סעיף א	1	2	3	4
סעיף ב	1	2	3	4	סעיף ב	1	2	3	4
סעיף ג	1	2	3	4	סעיף ג	1	2	3	4
סעיף ד	1	2	3	4	סעיף ד	1	2	3	4
סעיף ה	1	2	3	4	סעיף ה	1	2	3	4
סעיף ו	1	2	3	4	סעיף ו	1	2	3	4
סעיף ז	1	2	3	4	סעיף ז	1	2	3	4
סעיף ח	1	2	3	4	סעיף ח	1	2	3	4
סעיף ט	1	2	3	4					
סעיף י	1	2	3	4					

נספח ב': מילון מונחים
לשאלון 714911, אביב תשפ"ג

המונח	תרגום המונח		
	ערבית	רוסית	אנגלית
אחזור	استرجاع	Возврат, извлечение	retrieval
איבר	مُتَغَيِّر / عضو	Элемент	item
אקראי	عشوائي	Случайный	random
אתחול	قيمة بدائية	Инициализация	initialization
זמן ריצה	مدّة التنفيذ	Время работы	run time
טבלת ערבול (גיבוב)	جدول الخلط	Хеш-таблица	hash table
טיפוס	نوع	Тип	type
מחסנית	باغة	Стек	stack
מטריצת סמיכויות	جدول الحدود الزميلة	Матрица смежности	adjacency matrix
מיון טופולוגי	تصنيف	Топологическая сортировка	topological sorting
מסלול	مسار	Путь	path
מערך דינמי	مصفوفة غير ثابتة	Динамический массив	dynamic array
מצביע	مُؤَشِّر	Указатель	pointer
משתנה גלובלי	مُتَغَيِّر عامّ	Глобальная переменная	global variable
סדרה	سلسلة	Последовательность	series
סיבוכיות	تعقيد	Сложность (вычислений)	complexity
עדיפות	أولوية	Приоритет	preference
עץ חיפוש בינארי מאוזן	شجرة بحث ثنائي متوازنة	сбалансированное двоичное дерево поиска	balanced binary search tree
ערך מוחלט	قيمة مُطلَقة	модуль	absolute value
ערימה	كومة	Куча	heap
ערימה בינארית	كومة ثنائية	Двоичная куча	binary heap
פונקציית רקורסיבית	دالة أو عملية تراجعية	Рекурсивная функция	recursive function
צומת	مُفْتَرَق	Узел	node
קודקוד	رأس	вершина	vertex
קשת	وصلة	Дуга	arc
רשומה	سجِل	Запись (элемент структуры данных)	record
שדה קישור	حقل رابط	Поле, содержащее ссылку	linking field
שורש	جذر	Корень	root
תת-עץ	شجرة فرعية	Поддерево	sub-tree