714911 1

משרד החינוך

המִנהל הפדגוגי — אגף בכיר בחינות

מחברת בחינה

לנבחנים ולנבחנות שלום,

יש לקרוא את ההוראות בעמוד זה ולמלא אותן בדייקנות. אי־מילוי ההוראות עלול לגרום לתקלות ואף להביא לידי פסילת הבחינה.

הבחינה נועדה לבדוק הישגים אישיים, ולכן יש לעבוד עבודה עצמית בלבד. בזמן הבחינה אין להיעזר בזולת ואין לתת או לקבל חומר בכתב או בעל פה.

אין להכניס לחדר הבחינה חומר עזר — ספרים, מחברות, רשימות — חוץ מ"חומר עזר מותר בשימוש" המפורט בגוף השאלון או בהוראות מוקדמות של המשרד. כמו כן **אין להכניס לחדר הבחינה טלפונים או מכשירים אלקטרוניים אחרים**. שימוש בחומר עזר שאינו מותר יוביל לפסילת הבחינה.

לאחר סיום כתיבת הבחינה יש למסור את המחברת למשגיח ולעזוב בשקט את חדר הבחינה.

יש להקפיד על טוהר הבחינות !

הוראות לבחינה

1. יש לוודא כי במדבקות הנבחן שקיבלת מודפסים הפרטים האישיים שלך. אין להוסיף או לשנות שום פרט במדבקות, כדי למנוע עיכוב בזיהוי המחברת וברישום הציונים.
2. אם לא קיבלת מדבקה, יש למלא בכתב יד את הפרטים במקום המיועד למדבקת הנבחן.
3. אסור לכתוב בשולי המחברת (החלק המקווקו) משום שחלק זה לא ייסרק.
4. לטיוטה ישמשו אך ורק דפי מחברת הבחינה שיועדו לכך.
5. **אין לתלוש או להוסיף דפים**. מחברת שתוגש לא שלמה תעורר חשד לאי־קיום טוהר הבחינות.
6. אין לכתוב שם בתוך המחברת משום שהבחינה נבדקת בעילום שם.

ב ה צ ל ח ה !

מדבקות לנבחן
ملصقة ممتحَن

מדבקת נבחן והתאמות
ملصَقة ممتحَن وملاءَمات

שנה السنة חודש الشهر
מועד موعد

סמל ביה"ס رقم المدرسة
מס' תעודת הזהות رقم الهويّة

יש להדביק כאן ↑ מדבקת נבחן (ללא שם)
يجب هنا ↑ إلصاق ملصقة ممتحَن (بدون اسم)

מדבקת שאלון
ملصَقة نموذج امتحان

יש לסמן במשבצת ☐ אם ניתנה מחברת נוספת
يجب الإشارة في المربّع إذا أُعطي دفتر إضافيّ

* التّعليمات باللّغة العربيّة على ظهر الصّفحة

2

وزارة التّربية والتّعليم

الإدارة التّربويّة
للامتحانات

القسم الكبير

دفتر امتحان

تحيّة للممتحَنين وللممتحَنات،
يجب قراءة التّعليمات في هذه الصّفحة والعمل وفقًا لها بدقّة. عدم تنفيذ التّعليمات قد يؤدّي إلى عوائق مختلفة وحتّى إلى إلغاء الامتحان. أُعدّ الامتحان لفحص التّحصيلات الشّخصيّة، لذلك يجب العمل بشكل ذاتيّ فقط. أثناء الامتحان، لا يُسمح طلب المساعدة من الغير، ولا يُسمح إعطاء أو الحصول على موادّ مكتوبة أو شفهيّة.
لا يُسمح إدخال موادّ مساعدة – كتب، دفاتر، قوائم – إلى غرفة الامتحان، ما عدا "موادّ مساعدة يُسمح استعمالها" المفصّلة في نموذج الامتحان أو في تعليمات مسبقة من وزارة التربية والتعليم. كما **لا يُسمح إدخال هواتف خلويّة أو أجهزة إلكترونيّة أخرى إلى غرفة الامتحان**. استعمال موادّ مساعدة لا يُسمح استعمالها سوف يؤدّي إلى إلغاء الامتحان.
بعد الانتهاء من كتابة الامتحان، يجب تسليم الدّفتر للمراقب ومغادرة غرفة الامتحان بهدوء.

يجب التّقيّد بنزاهة الامتحانات !

تعليمات للامتحان

1. يجب التّأكّد بأنّ تفاصيلك الشّخصيّة مطبوعة على ملصَقات الممتحَن الّتي حصلت عليها. لا يُسمح إضافة أو تغيير أيّة تفاصيل في الملصَقات، وذلك لمنع عوائق في تشخيص الدّفتر وفي تسجيل العلامات.
2. في حال عدم حصولك على ملصَقة، يجب ملء التّفاصيل في المكان المعدّ لملصَقة الممتحَن، بخطّ يد.
3. لا يُسمح الكتابة في هوامش الدّفتر (في المنطقة المخطّطة)، لأنّه لن يتمّ مسح ضوئيّ لهذه المنطقة.
4. للمسوّدة تُستعمَل أوراق دفتر الامتحان المعدّة لذلك فقط.
5. **يُمنع نزع أو إضافة أوراق**. الدّفتر الّذي يُسلَّم ناقصًا يُثير الشّكّ بعدم الالتزام بنزاهة الامتحانات.
6. لا يُسمح كتابة الاسم داخل الدّفتر، لأنّ الامتحان يُفحص بدون ذكر اسم.

نتمنّى لكم النّجاح!

מדינת ישראל
משרד החינוך

סוג הבחינה: גמר לבתי־ספר לטכנאים ולהנדסאים
מועד הבחינה: אביב תשפ"ד, 2024
סמל השאלון: 714911
נספח: מילון מונחים

מבני נתונים ויעילות אלגוריתמים

הוראות לנבחנים

א. משך הבחינה: ארבע שעות ו־45 דקות.

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. כתבו את כל התשובות **בגוף השאלון**, במקומות המיועדים לכך.
2. עמודים 26–27 משמשים לטיוטה. כתיבת טיוטה בגוף השאלון או על דפים שמחוץ לשאלון עלולה לגרום לפסילת הבחינה.
3. ענו על מספר השאלות הנדרש בשאלון. המעריך יקרא ויעריך את מספר התשובות הנדרש בלבד, לפי סדר כתיבתן, ולא יתייחס לתשובות נוספות.
4. בנספח לשאלון זה מובא מילון מונחים בשפות עברית, ערבית, אנגלית ורוסית. תוכלו להיעזר בו בעת הצורך.

שאלון זה משמש כמחברת בחינה. הדביקו את מדבקת הנבחן במקום המיועד לכך בעמוד 1 (כריכה קדמית).

בשאלון זה 28 עמודים ועמוד אחד של נספח.

השאלות בשאלון זה מנוסחות בלשון רבים,
אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

המשך מעבר לדף

השאלות

בשאלון זה שני פרקים ובהם חמש שאלות. ענו על שני הפרקים, על־פי ההנחיות.

פרק ראשון (40 נקודות)

שאלה 1 – שאלת חובה (40 נקודות)

ענו על כל הסעיפים (א'–ד').

חברת התיירות 'מסביב לעולם' עוסקת בשיווק ובמכירה של טיסות מנתב"ג ליעדים שונים בעולם. החברה מעוניינת בתוכנה לניהול נתוני הטיסות. לפניכם הנתונים שינוהלו על ידי התוכנה: מספר הטיסה, היעד של כל טיסה, משך זמן הטיסה, מחיר הטיסה וחברת התעופה.

אבנר הציע לארגן את המידע באופן הזה:

פרטי הטיסות יישמרו בטבלת גיבוב על־פי מספר הטיסה. טבלת הגיבוב תצביע על מערך רשומות, שבו יופיעו הפרטים של כל טיסה: מספר הטיסה, מחיר הטיסה, משך זמן הטיסה, חברת התעופה ויעד הטיסה.

פונקציית הגיבוב (hash) תחזיר, עבור כל מספר טיסה (מפתח בטבלת גיבוב), את האינדקס המתאים לו בטבלת הגיבוב.

טבלת הגיבוב תהיה מסודרת כך שלא יהיו התנגשויות.

נוסף לכך, הטיסות יסווגו בעצי AVL על־פי יעדים. עבור כל יעד יהיה עץ AVL יחיד, וכל צומת בעץ יכיל את מספר הטיסה. עצי ה־AVL יהיו ממוינים לפי מחיר הטיסה.

הגדרת המבנה (struct) של טיסה:

```
// Define the Flight struct with a flight number
struct Flight {
   int flightNumber;
   double price;
   double hours;
   char companyName[30];
   char target[30];
};
```

הגדרה של טבלת הגיבוב (hash table):

```
// Define the HashTable struct
struct HashTable {
   int size;
   struct Flight** table; // Array of Flight pointers
};
```

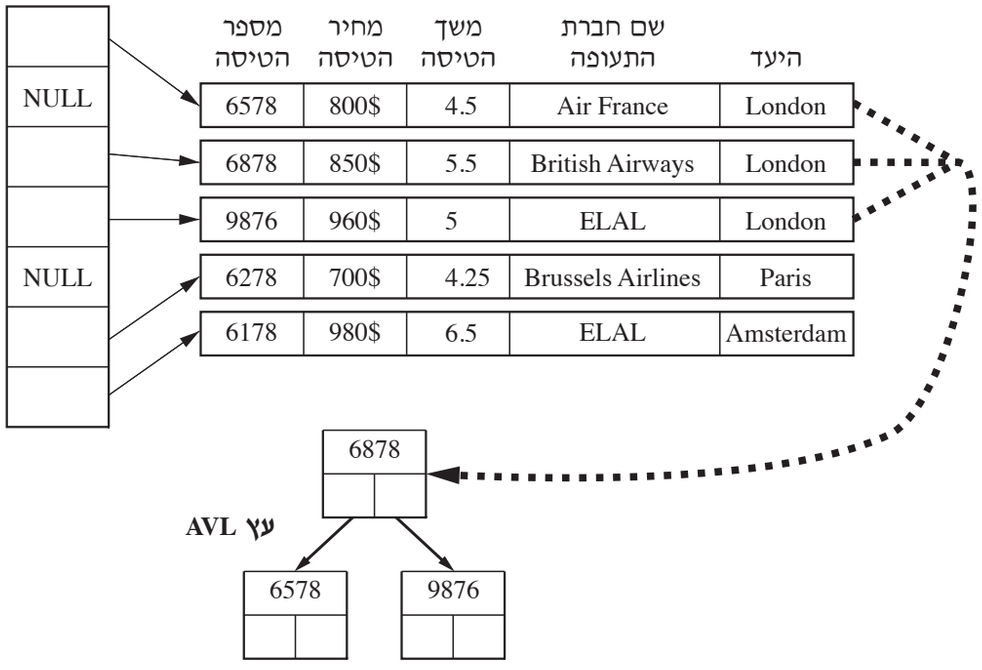
המשך בעמוד 5

אין לכתוב בשטח זה في هذه المنطقة لا تكتب

הגדרת המבנה (struct) של צומת בעץ AVL:

```
// Define the AVLTreeNode struct
struct AVLTreeNode {
   int flightNumber;
   struct AVLTreeNode* left;
   struct AVLTreeNode* right;
};
```

טבלת גיבוב



(10 נק') **א.** תייר מחפש טיסה ליעד מסוים. הוא מעוניין בטיסה הזולה ביותר, שאורכה אינו עולה על מספר שעות מסוים. לפניכם פונקצייה **רקורסיבית** למציאת טיסה מתאימה, שכותרתה:
findLeastPriceyFlightUnderHours

הפונקצייה מקבלת את עץ ה־AVL על־פי היעד המבוקש, ומחזירה את פרטי הטיסה הזולה ביותר שאורכה אינו עולה על maxHours שעות (משתנה המתקבל בקריאה לפונקצייה).

שימו לב, פונקצייה זו נעזרת בפונקצייה searchByFlightNumber שלהלן:

```
struct Flight* searchByFlightNumber(struct HashTable* hashTable, int flightNumber) {
   int index = hash(flightNumber, hashTable->size);
   return hashTable->table[index];
}
```

המשך בעמוד 6

אין לכתוב בשטח זה אלא לטיוטות בלבד. لا تكتب في هذه المنطقة

(8 נק') 1. השלימו את **ארבעת** הביטויים החסרים בפונקצייה זו:

```
struct Flight* findLeastPriceyFlightUnderHours(struct AVLTreeNode* root, double
maxHours, struct HashTable* hashTable) {
    struct Flight* flight = NULL;
    struct Flight* leftResult = NULL;

    if (root == NULL) {
        return NULL;
    }
    leftResult = findLeastPriceyFlightUnderHours (                    );
    if(leftResult != NULL) {
                                                                      ;
    }
flight = searchByFlightNumber(hashTable, root->flightNumber);
    if                                                                {
        return flight;
    }
    return findLeastPriceyFlightUnderHours (                          );
}
```

מפעילים את הפונקצייה findLeastPriceyFlightUnderHours על עץ AVL שיש לו n צמתים.

(1 נק') 2. מהי סיבוכיות זמן הריצה של הפונקצייה **במקרה הטוב ביותר**?

(1 נק') 3. מהי סיבוכיות זמן הריצה של הפונקצייה **במקרה הגרוע ביותר**?

המשך בעמוד 7

אין לכתוב באזור זה לא تكتب في هذه المنطقة

ב. (14 נק') תייר מחפש טיסה ליעד מסוים. הוא מעוניין בטיסה הקצרה ביותר, שמחירה אינו עולה על סכום מסוים. לפניכם פונקצייה **רקורסיבית** למציאת טיסה מתאימה, שכותרתה:
findShortestFlightUnderPrice

הפונקצייה מקבלת את עץ ה־AVL על־פי היעד המבוקש, ומחזירה את פרטי הטיסה שמשך הזמן שלה הוא הקצר ביותר ומחירה המְרַבי הוא maxPrice.

שימו לב, פונקצייה זו נעזרת בפונקצייה searchByFlightNumber שהוצגה קודם (בעמוד 5).

1. (12 נק') השלימו את **ששת** הביטויים החסרים בפונקצייה זו:

```
struct Flight* findShortestFlightUnderPrice(struct AVLTreeNode* root, int maxPrice,
struct HashTable* hashTable) {
    struct Flight* flight = NULL;
    struct Flight* leftResult = NULL;
    struct Flight* rightResult = NULL;

    if (root == NULL) {
        return NULL;
    }
    leftResult = findShortestFlightUnderPrice(                              );

    flight = searchByFlightNumber(hashTable, root->flightNumber);

    if (flight->price > maxPrice) {

                                                                            ;
    }

    if (                                                                    ) {
          flight = leftResult;
    }

    rightResult = findShortestFlightUnderPrice (                            );

    if                                                                      {
       return flight;
    }
                                                                            ;
}
```

מפעילים את הפונקצייה findShortestFlightUnderPrice על עץ AVL בעל n צמתים.

2. (1 נק') מהי סיבוכיות זמן הריצה של הפונקצייה **במקרה הטוב ביותר**? _____

3. (1 נק') מהי סיבוכיות זמן הריצה של הפונקצייה **במקרה הגרוע ביותר**? _____

המשך בעמוד 8

אין לכתוב באזור זה לא تكتب في هذه المنطقة
אין לכתוב באזור זה לא تكتب في هذه المنطقة

ג. (8 נק') ענת הציעה פתרון שונה מהפתרון של אבנר. היא מסכימה עם אבנר שנכון להשתמש בטבלת הגיבוב, אך חושבת שלא נכון להשתמש בעצי AVL.

לדעתה של ענת יש לבנות ערֵמת מינימום, שבה יישמרו הפרטים של כל הטיסות ליעד מסוים לפי מחיר טיסה. כל צומת בערֵמה יכיל את מספר הטיסה.

הניחו שמבנה הנתונים ישתנה על־פי הצעה זו, וציינו מהי סיבוכיות זמן הריצה למימוש כל אחת מן הפעולות שלפניכם.

1. (1.5 נק') חיפוש טיסה על־פי מחיר מינימלי ועד K שעות **במקרה הטוב ביותר**:

2. (1.5 נק') חיפוש טיסה על־פי מחיר מינימלי ועד K שעות **במקרה הגרוע ביותר**:

3. (1.5 נק') חיפוש טיסה על־פי משך זמן הטיסה המינימלי ועד מחיר S דולרים **במקרה הטוב ביותר**:

4. (1.5 נק') חיפוש טיסה על־פי משך זמן הטיסה המינימלי ועד מחיר S דולרים **במקרה הגרוע ביותר**:

5. (2 נק') הוספת טיסה חדשה למבנה הנתונים:

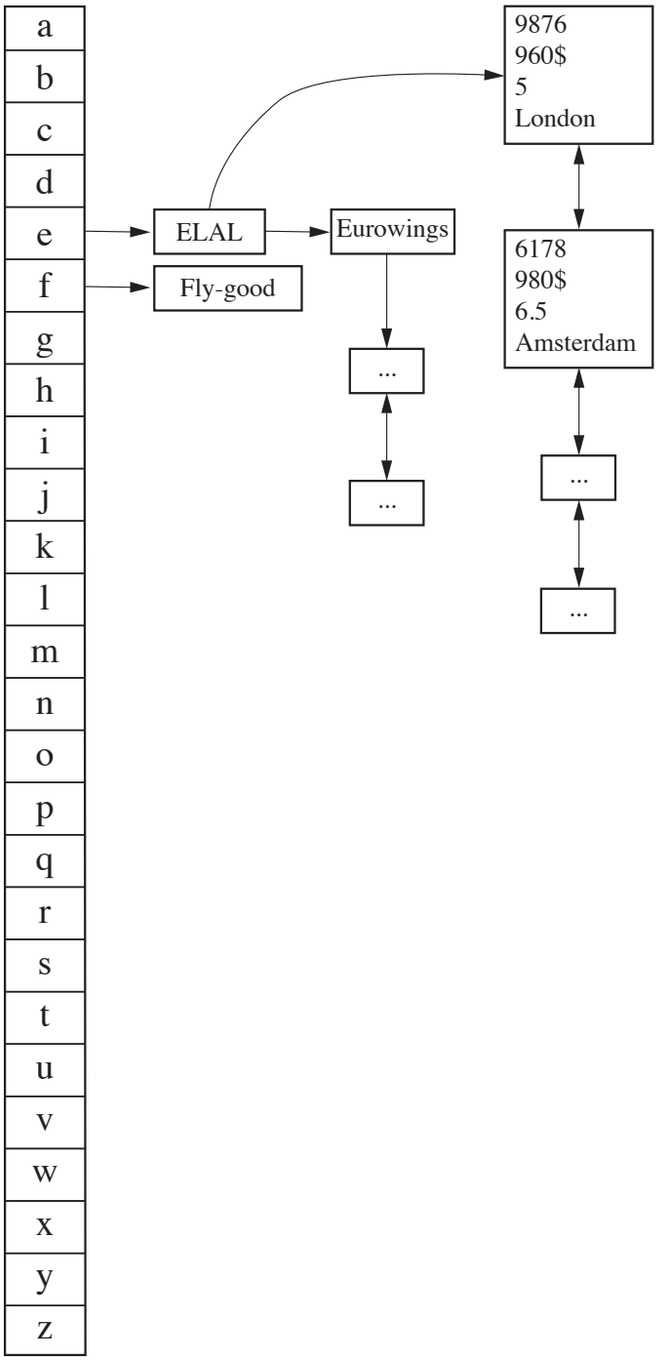
ד. (8 נק') חברת התיירות 'מסביב לעולם' שמרה את נתוני הטיסות במבנה נתונים שונה מזה שהציעו אבנר וענת. הנתונים נשמרו במערך בן 26 תאים לפי סדר ה־ABC. כל תא במערך מכיל מצביע על רשימה מקושרת חד־כיוונית, המכילה את השמות של חברות התעופה המתחילות באות השייכת לתא זה במערך (רשימה זו תיקרא להלן "רשימת החברות"). כל צומת ברשימת החברות מצביע על רשימה מקושרת דו־כיוונית של הטיסות היוצאות מנתב"ג והשייכות לאותה חברת תעופה (רשימה זו תיקרא להלן "רשימת הטיסות"). רשימת הטיסות ממוינת על־פי משך הטיסה, כאשר הטיסה הקצרה ביותר תופיע בתחילת הרשימה.

המשך בעמוד 9

אין לכתוב באזור זה / لا تكتب في هذه المنطقة

מבני נתונים ויעילות אלגוריתמים, אביב תשפ"ד, סמל 714911

לפניכם תרשים של מבנה נתונים זה:



המשך בעמוד 10

לפניכם הגדרת המערך:

```
CompanyFlight* arrFlight [26];
```

המבנה של כל צומת ברשימת החברות:

```
typedef struct CompanyFlight {
   char CompanyName[30];
   flight* nodeflight;
   CompanyFlight* next;
} CompanyFlight;
```

המבנה של כל צומת ברשימת הטיסות:

```
typedef struct flight {
   int FlightNumber;
   double price;
   double hours;
   char target[30];
   flight* prev;
   flight* next;
} flight;
```

(6 נק') 1. נתון שאורכה של כל רשימה מקושרת ב"רשימת החברות" אינו גדול מ־n ושאורכה של כל רשימה מקושרת ב"רשימת הטיסות" אינו גדול מ־m.
כתבו מהי סיבוכיות זמן הריצה של הפונקצייה, **במקרה הגרוע ביותר**, למימוש כל אחת מהפעולות שלפניכם.

I. הוספת טיסה חדשה למערכת:

II. חיפוש טיסה על־פי מספר הטיסה ושם החברה:

III. חיפוש טיסה על־פי שם החברה וזמן הטיסה המרבי (שם החברה וזמן הטיסה המרבי נתונים):

(2 נק') 2. אם רשימת הטיסות לא תהיה ממוינת, איזו פעולה מן הפעולות שבסעיף 1 שלעיל, יכולה להתבצע בסיבוכיות זמן ריצה טובה יותר (כלומר, קטנה יותר)?

המשך בעמוד 11

אין לכתוב באזור זה في هذه المنطقة لا تكتب

פרק שני (60 נקודות)

ענו על שלוש מן השאלות 2–5 (לכל שאלה – 20 נקודות).
בשאלות רב־ברירה, הקיפו בעיגול את הספרה המייצגת את התשובה הנכונה.

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:

$$T(n)=3T\left(\frac{n}{9}\right)+9\sqrt{n}\log n$$

מהי סיבוכיות זמן הריצה של האלגוריתם?

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

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

$$T(n) = 4T(n-1) - 4T(n-2) + 3n + 8$$

מהי סיבוכיות זמן הריצה של האלגוריתם?

1. $O(n)$
2. $O(n3^n)$
3. $O(n^3)$
4. $O(n2^n)$

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

$$T(n)=3T\left(\frac{n}{5}\right)+7\,n\log n$$

מהי סיבוכיות זמן הריצה של האלגוריתם?

1. $\Theta(n^{\log_5 3}\log n)$
2. $\Theta(n^{\log_5 3})$
3. $\Theta(n\log n)$
4. $\Theta(n^{\log_3 5}\log n)$

המשך בעמוד 12

אין לכתוב באזור זה / لا تكتب في هذه المنطقة

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

$$T(n) = 2T(\sqrt{n}) + 5\log n$$

מהי סיבוכיות זמן הריצה של האלגוריתם?

1. $\Theta(\sqrt{n}\log n)$
2. $\Theta(\log n \log\log n)$
3. $\Theta(\sqrt{n}\log^2 n)$
4. $\Theta(\log^5 \log n)$

סעיפים ה'–ז' מתייחסים לבעיה שלפניכם:

ברוב חברות ההיי־טק בישראל נהוג לפטר עובדים על־פי השיטה 'נכנס אחרון, יוצא ראשון'. חברת ההיי־טק 'היי סופט' חוששת להפסיד עובדים מבריקים בגלי פיטורין, רק מפני שהצטרפו אחרונים לחברה. לכן הוחלט שעובד חדש לא יפוטר בחמשת החודשים הראשונים לעבודתו בחברה. במהלך חודשים אלו יקבל העובד, מדי שבוע, ציון שייקבע בהתאם לטיב עבודתו ולהתרשמות מנהלו הישיר ממנו.

יינתן ציון בין 1 (הנמוך ביותר) ל־5 (הגבוה ביותר) לכל תחום מבין התחומים האלה: איכות קוד, עבודת צוות, עמידה ביעדים וחשיבה מחוץ לקופסה.

עובד נחשב לעובד חדש במשך חמשת החודשים הראשונים לעבודתו בחברה. בסיום תקופה זו מסכמים את הציון הכולל של העובד בכל התחומים.

אם הציון הכולל הוא פחות מ־340, העובד יוחזר לנוהל הרגיל 'נכנס אחרון, יוצא ראשון', הנהוג בכל חברה.

אם הציון הכולל הוא 340 ומעלה, העובד ייחשב עובד מצטיין. כל עובד מצטיין ימתין לקידום עד שיתפנה תפקיד בכיר יותר מהתפקיד שאותו הוא ממלא בהתאם לגובה הציון שקיבל.

ה. נתוני העובדים החדשים נשמרים בתור (queue). נוסף לכך, לצורך עדכון הציונים, נתוני העובדים החדשים נשמרים בטבלת גיבוב לפי מספר זהות.
מהו מבנה הנתונים המתאים והיעיל ביותר לאחסון נתוני יתר העובדים, לפי מדיניות הפיטורים והקידום של החברה?

1. מערך לעובדים מצטיינים, ותור לעובדים שאינם מצטיינים ואינם חדשים.
2. תור עדיפויות (priority queue) לעובדים מצטיינים, ומחסנית לעובדים שאינם מצטיינים ואינם חדשים.
3. תור עדיפויות (priority queue) לעובדים מצטיינים, ותור לעובדים שאינם מצטיינים ואינם חדשים.
4. מערך לעובדים מצטיינים, ומחסנית לעובדים שאינם מצטיינים ואינם חדשים.

ו. מהי סיבוכיות זמן הריצה לפיטורי עובד שאינו חדש ואינו מצטיין?

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

המשך בעמוד 13

الرجاء الكتابة في هذه المنطقة لا تكتب في هذه المنطقة

ז. מהי סיבוכיות זמן הריצה לעדכון ציון של עובד חדש?

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

סעיפים ח' ו־ט' מתייחסים לקטע הקוד שלפניכם:

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

ח. מהי הנוסחה הרקורסיבית עבור זמן הריצה של פונקצייה f?

1. $T(n)=T(n-1)+\frac{n}{2}$
2. $T(n)=T(n-1)+\log n$
3. $T(n)=T(n-1)+1$
4. $T(n)=T(n-1)+n^2$

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

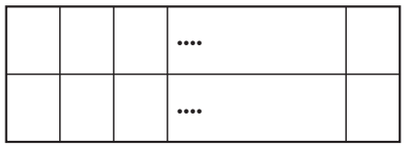
1. $\Theta(n)$
2. $\Theta(n\log n)$
3. $\Theta(n^2)$
4. $\Theta(n^3)$

אין לכתוב באזור זה לא تكتب في هذه المنطقة

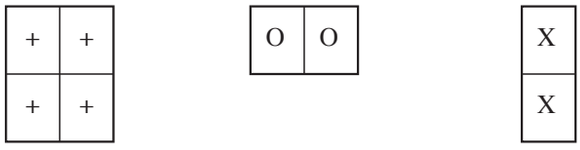
המשך בעמוד 14

ג. לפניכם מלבן בגודל $2\times n$ ומתחתיו שלוש צורות.

מלבן



שלוש צורות



$f(n)$ הוא מספר האפשרויות לכסות את המלבן בגודל $2\times n$ באמצעות שלוש הצורות.

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

המשך בעמוד 15

שאלה 3

ענו על <u>כל</u> הסעיפים א'–ה'.

נתונות סדרות בינאריות (של אפסים ואחדות) בגודל n כל אחת.
כדי להפיק מידע מהסדרות צריך לבצע פעולות של הכנסה וחיפוש.

א. (2 נק') מהו מבנה הנתונים המתאים כדי שפעולות אלו תתבצענה בזמן O(n)?

1. גרף מכוון קשיר חזק - כל סדרה בינארית נשמרת כמסלול בתוכו.
2. עץ בינארי – כל סדרה בינארית נשמרת כמסלול בעץ (בדומה לעץ הופמן).
3. מטריצה דלילה – כל סדרה בינארית נשמרת כרשימת סמיכויות.
4. רשימת רשימות – כל צומת מצביע על סדרה בינארית שנשמרת כרשימת סיביות.

ב. (4 נק') נתונה קבוצה של ארבע סדרות בינאריות: 00110 , 01010 , 01101 , 10101 .
כל סדרה היא בת 5 סיביות.

סרטטו את מבנה הנתונים שבו בחרתם בתשובה לסעיף א', כך שיתאים לקבוצה הנתונה בסעיף זה.

המשך בעמוד 16

לא לכתוב באזור זה
لا تكتب في هذه المنطقة

ג. (6 נק') לפניכם פונקציית הכנסה **רקורסיבית** שכותרתה: Insert. הפונקצייה מקבלת מצביע לעץ קיים וסדרה בינארית בגודל n>0. בהתאם לסדרה הבינארית מתקיים מסלול בעץ שמתחיל מהשורש שלו, כך שלכל '1' בסדרה המסלול פונה לצומת שהוא בן ימני, ולכל '0' המסלול פונה לצומת שהוא בן שמאלי. הסדרה הבינארית מיוצגת על-ידי מערך של תווים. כל תו במערך מקבל רק אפסים ואחדות.

הערה: הפונקצייה newNode() יוצרת צומת חדש, T, שבו T->right = T->left = NULL, ומחזירה מצביע עליו.

השלימו את **שלושת** הביטויים החסרים בפונקצייה זו:

```
1.  void Insert(struct Node* T, char* s, int n)
2.  {
3.      if (n == 0)
4.      {
5.          return;
6.      }
7.    [                                                    ]
8.      {
9.          if( T-> right == NULL)
10.             T->right = newNode();
11.       [                                                ]
12.     }
13      else
14.     {
15.         if( T->left== NULL)
16.             T->left = newNode();
17.       [                                                ]
18.     }
19. {
```

ד. (2 נק') מהי סיבוכיות זמן הריצה של הפונקצייה Insert?

1. O(logn)
2. O(n)
3. O(nlogn)
4. $O(n^2)$

המשך בעמוד 17

אין לכתוב באזור זה לא تكتب في هذه المنطقة

ה. (6 נק') כותבים פונקציית חיפוש על־סמך הפונקצייה Insert הנתונה בסעיף ג'.
כותרת הפונקצייה החדשה היא int Search(struct Node* T, char* s, int n).
אם המערך s מופיע בעץ T הפונקצייה תחזיר 1 , אחרת הפונקצייה תחזיר 0 .
בכל מקום שבו הופיעה הפקודה Insert בפונקצייה הקודמת, מחליפים אותה בפקודה Search בפונקצייה החדשה.
אילו מן השורות בפונקצייה הקודמת יימחקו ואילו יישארו או ישתנו בפונקצייה החדשה?
עבור כל שורה הקיפו אם נשארת, נמחקת או משתנה.
אם השורה משתנה, כתבו את שורת הקוד החדשה הנדרשת.
שימו לב, שורה 9 נתונה כ"נשארת".

שורה 3: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

שורה 5: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

שורה 7: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

שורה 9: (נשארת) / נמחקת / משתנה

שורה 10: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

שורה 11: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

שורה 15: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

שורה 16: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

שורה 17: נשארת / נמחקת / משתנה
אם סימנתם "משתנה", כתבו את שורת הקוד החדשה: _____

לא לכתוב באזור זה لا تكتب في هذه المنطقة

המשך בעמוד 18

שאלה 4

ענו על כל הסעיפים א'–ט'.

(2 נק') **א.** נתון קטע קוד:

```
for (i=1; i<2n; i++)
{
   x=1/i;
  x=sqrt(x);
}
a=n;
while(a>0)
  a-=x;
```

נוסף על כך נתון כי המשתנים a,x הם מטיפוס double, וסיבוכיות החישוב sqrt(x) היא O(1).

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

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

(2 נק') **ב.** נתונות ארבע סדרות המייצגות הדפסה במעבר **postorder** על עץ בינארי.

איזו סדרה **אינה** מייצגת עץ חיפוש בינארי?

1. 201, 154, 300, 100, 400, 66, 405
2. 50, 300, 154, 100, 400, 405, 201
3. 50, 100, 154, 300, 405, 400, 201
4. 100, 201, 154, 405, 400, 300, 50

(2 נק') **ג.** עבור הטקסט T מעל א"ב $\Sigma=\{a,b,c,d,e\}$, נתון כי:

- האות **a** מופיעה **6** פעמים, האות **b** מופיעה **5** פעמים והאות **e** מופיעה **6** פעמים.
- קוד הופמן של האותיות הוא: $a-10,\ b-00,\ c-010,\ d-011,\ e-11$.
- מספר ההופעות של תת־עץ שמאלי **אינו** גדול יותר ממספר ההופעות של תת־עץ ימני.

הניחו שסיבית הקוד לצומת הימני היא 1 ולשמאלי 0 .

כתבו את כל האפשרויות לשכיחות האותיות **c** ו־**d** בטקסט T.

המשך בעמוד 19

الرجاء الكتابة في هذه المساحة فقط. لا تكتب في هذه المساحة

סעיפים ד'–ו' מתייחסים לבעיה שלפניכם:

עץ "משפחה" הוא עץ שנבנה על־סמך סדרת אברים, ובו כל איבר במקום ה־i הוא בן של האיבר במקום $\frac{i}{2}$, כאשר בין שני בנים לאותו אב קיים סדר כך שהקטן יותר הוא בן שמאלי, והגדול יותר הוא בן ימני.

הערה: מקום האיבר הראשון הוא i=1 .

ד. **(4 נק')** סרטטו את העץ "משפחה" לסדרת האיברים הזאת (קראו משמאל לימין):
16, 10, 30, 15, 27, 6, 40, 50, 3 .

ה. **(2 נק')** אם סדרת האיברים המוכנסים לעץ "משפחה" ממוינת בסדר עולה, מה מהבאים **לא** יקרה?

1. נקבל עץ שהסריקה שלו לפי רמות תהיה זהה לסדרה.
2. נקבל עץ חיפוש בינארי.
3. נקבל עֵרמת מינימום.
4. סדר הבנים בעץ יהיה כסדרם בסדרה.

המשך בעמוד 20

לא לכתוב באזור זה
لا تكتب في هذه المنطقة

(2 נק') **1.** מהי סיבוכיות זמן הריצה של חיפוש איבר בעץ "משפחה"?

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

סעיפים ז'–ט' מתייחסים לבעיה שלפניכם:

נתון מערך A של n נקודות במישור (x_i,y_i) . $1 \le i \le n$

נדרש לבדוק אם קיים ישר העובר דרך ראשית הצירים, כך שלפחות 35% מ־n הנקודות הנתונות נמצאות עליו.

לפניכם פסאודו־קוד (קוד מדומה) של אלגוריתם שנועד לבדוק אם קיים ישר כזה:

```
Algo_line (A,n)
//D array of size n

for i=1 to n
  D[i]= yi/xi;
k= _____(1)_____;
count=0;
for i=1 to n
  if(D[i]==k)
     count++;
if (count>=0.35*n)
     return TRUE;
k= _____(2)_____;
count=0;
for i=1 to n
  if(D[i]==k)
      count++;
if (count>=0.35*n)
     return TRUE;
return FALSE;
```

המשך בעמוד 21

(2 נק') **ז.** מהו הביטוי החסר המיוצג על־ידי הספרה 1 ?

1. Select(D,1);
2. Select(D,n/3);
3. Select(D,n/2);
4. Select(D,n);

(2 נק') **ח.** מהו הביטוי החסר המיוצג על־ידי הספרה 2 ?

1. Select(D,n/2);
2. Select(D,n/3);
3. Select(D,2n/3);
4. Select(D,n);

(2 נק') **ט.** מהי סיבוכיות זמן הריצה של האלגוריתם Algo_line?

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

لا تكتب في هذه المنطقة
אין לכתוב באזור זה

המשך בעמוד 22

שאלה 5

ענו על כל הסעיפים א'–ז'.

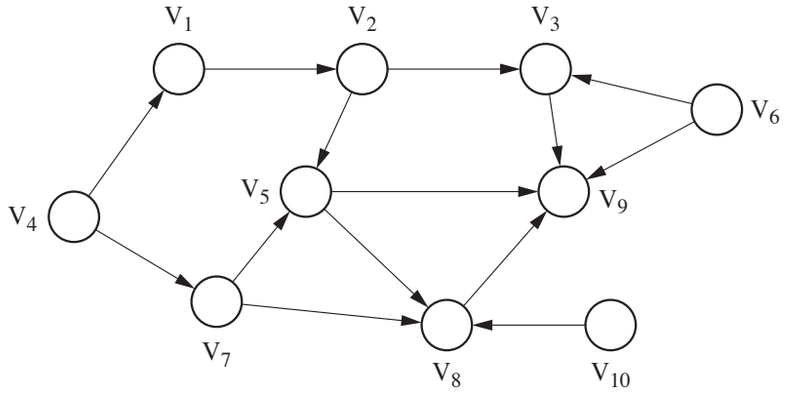
סעיפים א'–ה' מתייחסים לבעיה שלפניכם:

לפניכם פסאודו־קוד (קוד מדומה) של אלגוריתם ALGO המקבל גרף **מכוון** G = (V,E) בייצוג של רשימות שכנים.

```
ALGO(G=(V,E))
    count=0
    //D array of size |V|
    for each u∈ V
          D[u] = 0
    for each u∈ V
          for each v∈ adj(u) // רשימת שכנים
                D[v]++
//Q queue
      for each u∈ V
            if (D[u] = 0)
                  enqueue(Q,u)
      while (not empty(Q))
            u = dequeue(Q)
            print(u)
            count++
            for each v∈ adj(u)
                  D[v]--
                  if D[v] = 0
                        enqueue(Q,v)
if (count < |V|)
      return FALSE
return TRUE
```

המשך בעמוד 23

(4 נק') **א.** מריצים את האלגוריתם על הגרף הזה:

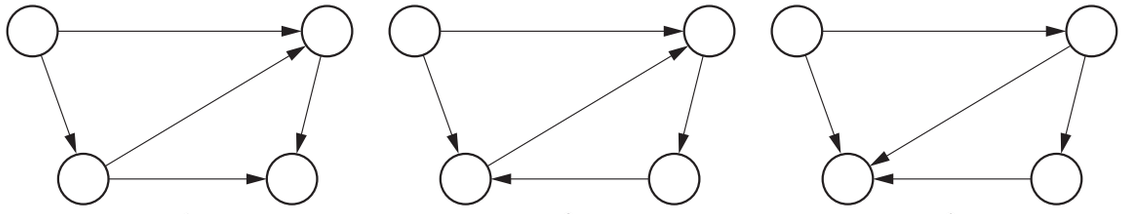


איזו סדרה **לא** תודפס על־ידי האלגוריתם באופן ודאי (קראו משמאל לימין)?

1. v4, v10, v6, v1, v7, v2, v3, v5, v8, v9
2. v4, v10, v6, v7, v1, v2, v5, v3, v8, v9
3. v4, v6, v10, v7, v1, v5, v8, v2, v3, v9
4. v4, v6, v10, v7, v1, v2, v5, v3, v8, v9

(2 נק') **ב.** איזו תשובה האלגוריתם יחזיר עבור הגרף המופיע בסעיף א'?

(2 נק') **ג.** לאיזה גרף, מן הגרפים שלפניכם, האלגוריתם יחזיר FALSE?
הקיפו את התשובה הנכונה.



(2 נק') **ד.** כיצד קוראים לסדר הצמתים שהאלגוריתם **מדפיס**?

(2 נק') **ה.** מהי סיבוכיות זמן הריצה של האלגוריתם?

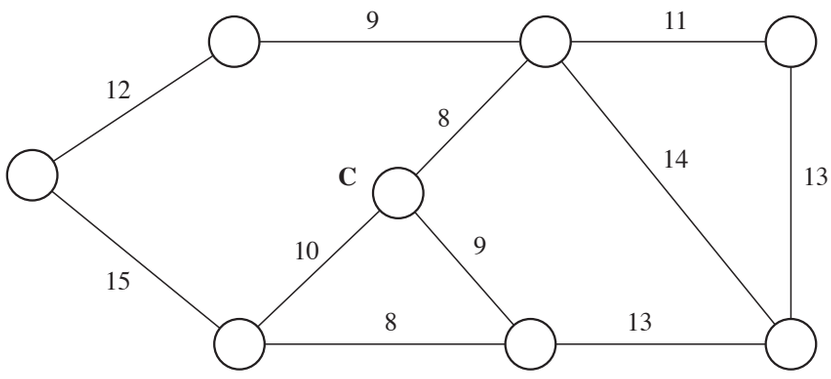
1. $\Theta(|V|^2)$
2. $\Theta(|V|+|E|)$
3. $\Theta(|V|)$
4. $\Theta(|V|*|E|)$

המשך בעמוד 24

לא לכתוב באזור זה لا تكتب في هذه المنطقة
لا تكتب في هذه المنطقة لا לכתוב באזור זה

סעיפים ו'–ז' אינם קשורים לסעיפים הקודמים.

(4 נק') **ו.** נתון גרף:



מריצים את האלגוריתם PRIM על הגרף החל מקודקוד C.

לפניכם שתי טענות:

A: סדר **משקלים** אפשרי של צלעות העץ ש־**PRIM** הכניס לעפ"מ (עץ פורש מינימלי) הוא (משמאל לימין):

8, 9, 8, 9, 11, 12, 13

B: סדר **משקלים** אפשרי של צלעות העץ ש־**PRIM** הכניס לעפ"מ הוא (משמאל לימין):

8, 9, 9, 8, 11, 12, 13

לפניכם ארבע טענות. מהי הטענה הנכונה מביניהן?

1. טענה A נכונה, וטענה B אינה נכונה.
2. טענה B נכונה, וטענה A אינה נכונה.
3. הטענות A ו־B נכונות.
4. הטענות A ו־B אינן נכונות.

המשך בעמוד 25

(4 נק') **ז.** לפניכם טבלת שכיחויות.

האות	השכיחות
a	65
b	15
h	10
l	13
m	50
o	20
s	25

קודדו את המילה shalom לפי קוד הופמן הבנוי בהתאם לטבלה זו.

אין לכתוב באזור זה אלא בטיוטה. لا تكتب في هذه المنطقة

אין לכתוב באזור זה אלא בטיוטה. لا تكتب في هذه المنطقة

בהצלחה!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם אלא ברשות משרד החינוך.

מבני נתונים ויעילות אלגוריתמים, אביב תשפ"ד, סמל 714911

טיוטה

لا تكتب في هذه المنطقة لא לכתוב באזור זה

מבני נתונים ויעילות אלגוריתמים, אביב תשפ"ד, סמל 714911

טיוטה

לא לכתוב באזור זה
لا تكتب في هذه المنطقة

מדבקת משגיח
ملصقة مراقب

"איתך בכל מקום, גם בבגרות.
בהצלחה, מועצת התלמידים והנוער הארצית"
"معك في كلّ مكان، وفي البجروت أيضًا.
بالنّجاح، مجلس الطّلّاب والشّبيبة القطريّ"

נספח: מילון מונחים

לשאלון 714911, אביב תשפ״ד

המונח	תרגום המונח		
	ערבית	רוסית	אנגלית
אחזור	استرجاع	Возврат, извлечение	retrieval
איבר	مُتَغيِّر/عضو	Элемент	item
אקראי	عشوائي	Случайный	random
אתחול	قيمة بدائيّة	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	مُتَغيِّر عامّ	Глобальная переменная	global variable
סדרה	سلسلة	Последовательность	series
סיבוכיות	تعقيد	Сложность (вычислений)	complexity
עדיפות	أولويّة	Приоритет	preference
עץ חיפוש בינארי מאוזן	شجرة بحث ثنائيّ متوازنة	сбалансированное двоичное дерево поиска	balanced binary search tree
ערך מוחלט	قيمة مُطْلَقة	модуль	absolute value
ערימה	كومة	Куча	heap
ערימה בינארית	كومة ثنائيّة	Двоичная куча	binary heap
פונקציית רקורסיבית	دالّة أو عمليّة تراجعيّة	Рекурсивная функция	recursive function
צומת	مُفْتَرَق	Узел	node
קודקוד	رأس	вершина	vertex
קשת	وصلة	Дуга	arc
רשומה	سِجِلّ	Запись (элемент структуры данных)	record
שדה קישור	حقل رابط	Поле, содержащее ссылку	linking field
שורש	جذر	Корень	root
תת־עץ	شجرة فرعيّة	Поддерево	sub-tree