משרד החינוך
המזכירות הפדגוגית
הפיקוח על הוראת מדעי המחשב

מכון הנרייטה סאלד
המרכז לבחינות בגרות

<u>תדריך לבדיקת בחינות הבגרות במדעי המחשב, 899271, קיץ תשפ"ו</u>

<u>כללי</u>:

1. על שגיאה בתחביר בשפת המחשב להוריד עד 3%.
2. על כל זימון של פעולה ללא ( ) להוריד 1%, ולכל היותר 3% לשאלה.
3. על שימוש שגוי בפעולת ממשק להוריד 5% לכל סוג פעולה, ולכל היותר 15% לשאלה.
4. על שימוש במשתנה שלא הוצהר/ לא אותחל/אותחל באופן שגוי, להוריד 2%.
5. על כל הצהרה שגויה של פרמטר (מערך ללא סוגרים וכדומה) להוריד 2%.
6. על שימוש בקיצורי הוראות במקום בהוראות מלאות, להוריד 5% פעם אחת לכל השאלון.
7. על שימוש בממשקים השייכים לתוכנית הלימודים הישנה או לממשקים בשפה, להוריד עד 10% לשאלה.
8. לולאה אינסופית – להוריד 15% על כל לולאה.
9. גישה לתכונות פרטיות (ללא get ו- set), להוריד 5% לטעות, ולכל היותר 10% לשאלה.
10. גישה שגויה לתכונות פרטיות (get במקום set, ולהיפך), להוריד 5% לטעות, ולכל היותר 10% לשאלה.
11. הדפסה במקום החזרה ולהיפך- להוריד 10%

שאלה 1
סעיף א – 55%

סעיף א(1) – 45%

פתרון המשתמש בתור יחיד ובזקיף, ללא תור עזר, ובלבד שהוא מבצע סריקה מלאה של התור ואיחוד תקין של ההזמנות -יזוכה במלוא הנקודות

- *חזרה עבור כל ההזמנות – 5%*
- *זיהוי כל ההזמנות של אותו לקוח – 10%*
- *צבירת count נכונה (או עדכון) - 10%*
- *מניעת כפילויות - 10%*
- *בניית התור המאוחד והחזרה - 10%*

– אם החזירו את התור המקורי לאחר עדכון או השתמשו בהזמנה קיימת ועדכנו אותה – לא להוריד

– השתמשו ב- head במקום remove - להוריד 5% פעם אחת
– גישה לערכים ללא get/set - להוריד 5% פעם אחת
– בדקו תור ריק בצורה שגויה (למשל null) – להוריד 5%
– בנו הזמנה חדשה בצורה שגויה/השתמשו בset בצורה שגויה – להוריד 5%
– אתחלו תור חדש בצורה שגויה - להוריד 5%
– סכמו בצורה שגויה (למשל, לא סכמו את ההזמנה הראשונה) – להוריד 5%
– שלפו ישירות לתוך תור עזר ובדקו head (לא שמרו קודם במשתנה זמני)– להוריד 5%
– עשו לולאת מספרים עד מספר גבוה שרירותי (לא קוד מקסימלי בתור ולא Integer מקסימלי) – להוריד 10%
– הכניסו לקוח לא קיים (בלולאה של מספרים הכניסו לקוח עם 0 מוצרים) – להוריד 10%
– לולאה אינסופית – להוריד 5%
– לא סכמו את כל ההזמנות של אותו לקוח – להוריד 10%
– לא זיהה את כל הלקוחות (לא הגדירו לולאה חיצונית) – להוריד 25%
– איבדו את התור לאחר טיפול (לגמרי או לאחר לקוח אחד) – להוריד 10%
– התייחסו לתור כרשימה (getValue, getNext) להוריד 20% או מערך [!] – להוריד 30%

סעיף א(2) – 10%

- יעילות n^2 (n מספר ההזמנות בתור qOrder) – 5%
- נימוק – 5%

- כתבו פרמטר מבלי להגדיר שהוא אורך התור qOrder - להוריד 3%

סעיף ב – 45%

סעיף ב(1) – 35%

- קבלת תור מאוחד (זימון הפעולה מסעיף א) – 5%
- לולאה על התור המאוחד – 5%
- שליפה הזמנה – 5%
- בדיקת כמות, והכנסה לתור מתאים – 5%
- איחוד התורים של שני סוגי הלקוחות בסדר הנכון – 10%
- החזרה – 5%

- הורדות זהות לסעיף א
- בדקו לפי הזמנה ספציפית (לא איחדו את ההזמנות) – להוריד 15%
- זימון שגוי (למשל כפעולה פנימית) – להוריד 5%
- אתחלו תור מטיפוס שלם ושמרו הזמנות במקום תז – להוריד 10%
- אתחלו תור מטיפוס הזמנה ושמרו הזמנות במקום ת״ז– להוריד 5%
- שגו בבדיקה (למשל, מעל 10) – להוריד 5%
- סדר הפוך (מועדפים למטה)– להוריד 5%

<u>סעיף ב(2) – 10%</u>

- יעילות n^2 (n מספר הזמנות בתור qOrder) – 5%
- נימוק (זימון הפעולה מסעיף א) – 5%

- כתבו ש- n הוא אורך התור המאוחד (או לא התחשבו בסיבוכיות סעיף א) - להוריד 5%

שאלה 2

סעיף א -50%

- לולאה ובדיקת שני תנאי עצירה (len+null) – 15%
- התקדמות בשתי השרשרות באורך הנדרש – 15%
- בדיקת פלוס/מינוס וסכימה – 15%
- החזרה – 5%

- העבירו את ערכי הרשימה למבנה אחר (מערך / תור ...) והפתרון תקין – לא להוריד
- פנייה ל null (חריגה מגבולות הרשימה באיבר אחד/ השתמשו ב isEmpty) - להוריד 5%
- הגדירו מצביע מטיפוס שגוי– להוריד 5%
- לא עברו על כל החוליות הנדרשות (פספסו ב -1) – להוריד 5%
- התייחסות שגויה לחוליה הראשונה (למשל, שיש פעולת חשבון לפניה) – להוריד 5%
- קידמו את מצביע הפעולה באופן שגוי (למשל, גם עבור החוליה הראשונה) – להוריד 5%
- דילגו על ערכים (קידום כפול) – להוריד 5%
- גישה לחוליה/לערכים ללא get - להוריד 5% פעם אחת
- לא התייחסו בכלל ל – len (חישוב עד סוף השרשרת) או לא התייחסו ל null - להוריד 15%
- תנאי שגוי or במקום and – להוריד 5%
- אם השתמשו ב while ולא עדכנו len - להוריד 5%
- פספסו את len באחד – להוריד 5%
- לא קידמו את המצביע (עשו לולאה אך שכחו לקדם מצביע) – להוריד 5% על כל שרשרת
- החזירו ערך בתוך לולאה (במקום בסוף הלולאה) – להוריד 10%
- החזירו ערך תקין בתוך לולאה ורק שגיאת קומפילציה (לא הוסיפו בסוף return) להוריד 5%
- בדיקה שגויה של פלוס/מינוס – להוריד 5%
- איפסו משתנה סוכם בתוך הלולאה – להוריד 10%
- הרס שרשרת – להוריד 10%

סעיף ב – 50%

- בדיקת תנאי עצירה מתאים – 10%
- התקדמות בשתי השרשרות – 10%
- זימון הפעולה מסעיף א – 10%
- בדיקה מול val – 10%
- החזרה true אם שווה ל val – 5%
- החזרה false מחוץ לולאה – 5%

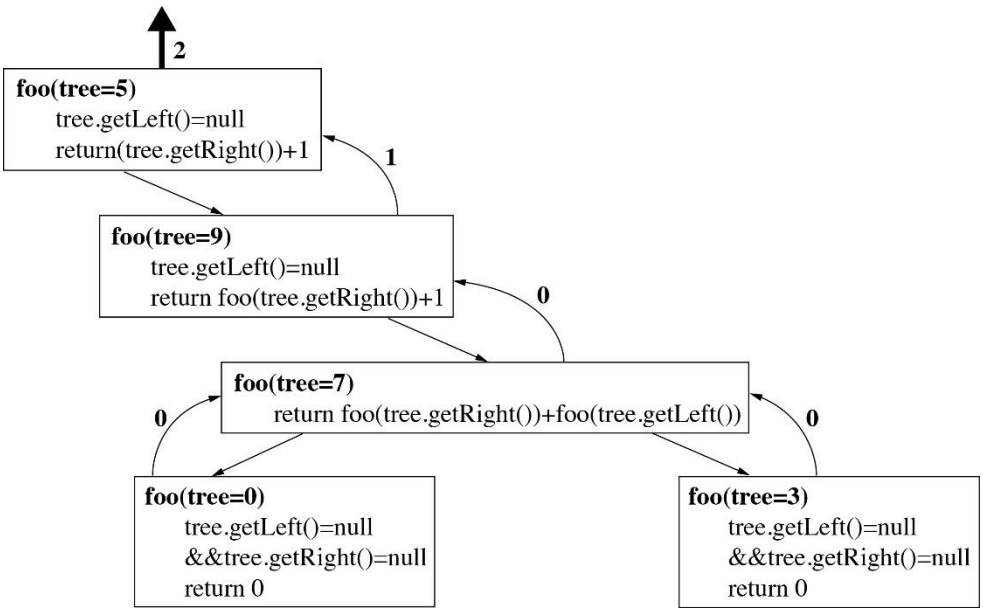
- הורדות זהות לסעיף א
- החזרה על פי החוליה האחרונה (עדכון true/false בכל חוליה) – להוריד 10%
- החזרה false בתוך הלולאה – להוריד 10%
- זימון שגוי (כפעולה פנימית או רק שרשרת אחת או סדר שגוי) – להוריד 5%
- חריגה מרשימה – להוריד 5%
- פספסו את המספר האחרון בשרשרת המספרים – להוריד 5%
- קידמו רק את שרשרת המספרים - להוריד 15%
- כתבו שוב את הפעולה מסעיף א –לא להוריד על אותם השגיאות שכבר ירדו קודם
- אם שלחו לסעיף א' רשימה ריקה של מספרים, ללא טיפול ב null בסעיף א' – להוריד 5%

שאלה 3

סעיף א – 60%

סעיף 1 – 15%

- מעקב – 15% (חמישה צמתים – כל צומת 3%)

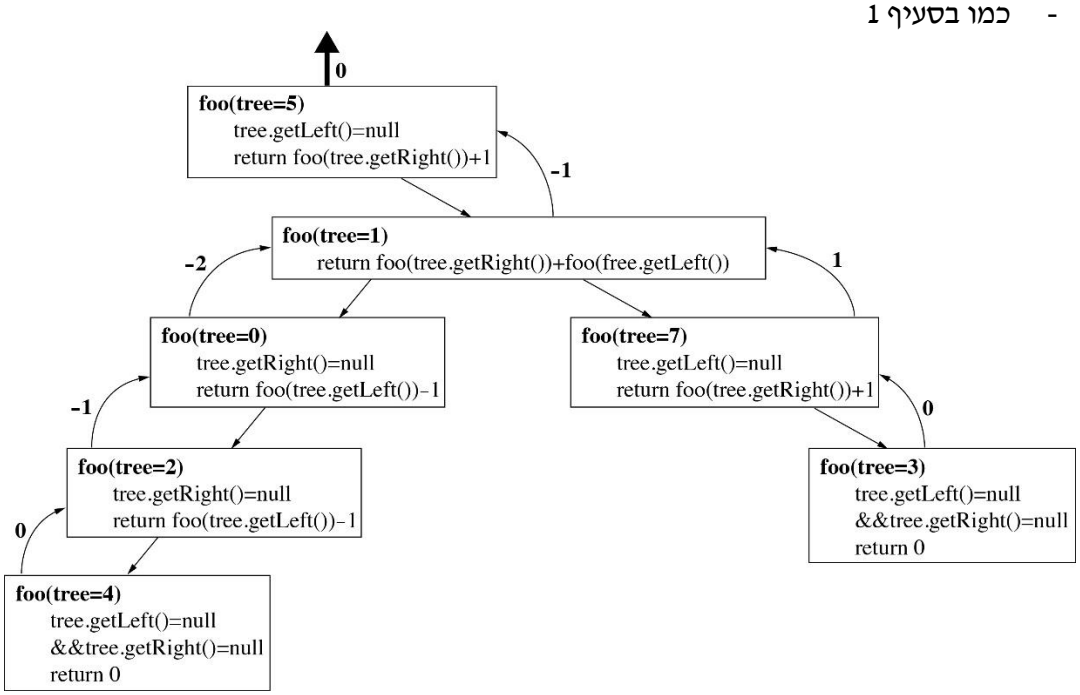


- טעו במספר המוחזר – יש להוריד פעם אחת והמשך בדיקה לשיטתו
- אם אין מעקב בכלל– **לא לתת נקודות על הסעיף**
- אם לא הראה תהליך חזרה עם ערך מוחזר בכל זימון – להוריד 5% סה"כ

<u>סעיף 2 – 21%</u>

- מעקב – 21% (7 צמתים – כל צומת 3%)

  - כמו בסעיף 1



<u>סעיף 3 – 8%</u>

- הערך הגבוה ביותר (5) – 4%+ הצגה (עץ שרשרת (שרוך) ימינה עם 6 צמתים) – 4%

  - פספסו באחד (כתבו 6 או שרטטו 5 או 7 צמתים) – להוריד 2%
  - שגה בכיוון העץ – להוריד 2%

<u>סעיף 4 – 8%</u>

- הערך הנמוך ביותר (-5) – 4%+ הצגה (עץ שרשרת (שרוך) שמאלה עם 6 צמתים) – 4%

  - פספס באחד (כתבו -6 או שרטטו 5 או 7 צמתים) – להוריד 2%
  - שגה בכיוון העץ – להוריד 2%

<u>סעיף 5 – 8%</u>

- ההפרש בין מספר הקשתות (הבנים) הימניים לשמאליים (מספר בנים ימניים פחות ממספר בנים שמאליים) – 8%

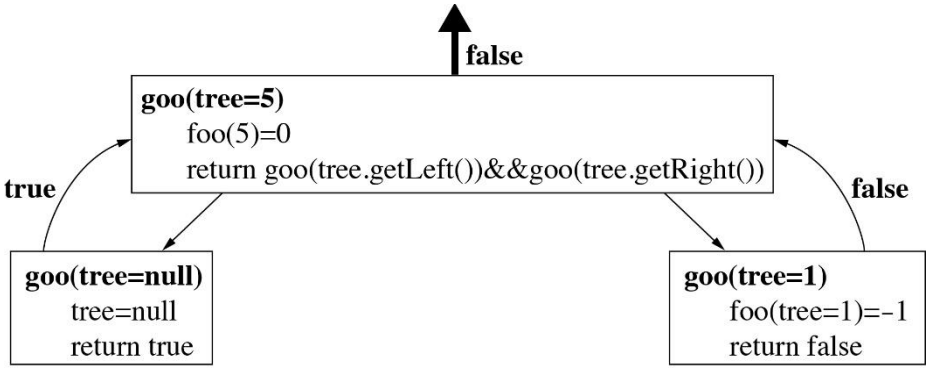
  - כל ניסוח דומה יתקבל
  - תיארו את אופן החישוב ולא את המטרה לא לתת נקודות על הסעיף

סעיף ב – 40%

סעיף ב 1 – 25%

- מעקב – 25% :
  - שורש – 11% : foo(5) =0 – 5% + פנייה והחזרה משני הבנים – 6%
  - בן שמאלי – 6% : tree=null – 3% + return true – 3%
  - בן ימני – 8% : foo(1)!=0 – 5% + return false – 3%

- אם אין מעקב בכלל– **לא לתת נקודות על הסעיף**



סעיף ב 2 – 15%

- הצגת עץ (עץ מלא של 7 חוליות או 5 חוליות ללא בנים יחידים) – 15% כל עץ בינארי מלא, לפחות 5 צמתים (ללא צמתים בעלי בן יחיד) יתקבל.

- אם טעו בסעיפים קודמים, לבדוק לשיטתו
- אם שרטטו פחות מ 5 צמתים – לא לתת ניקוד

אלגוריתמים

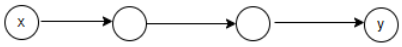
שאלה 4

סעיף א – 60%

- כל סעיף 12% (4% תשובה ו-8% נימוק/הפרכה) כל דוגמה לפחות 4 צמתים
- סעיף 1 – נכון. הקשת היא דו כיוונית ולכן אפשר להגיע לכל הצמתים ברכיב הקשיר.
- סעיף 2 –נכון. בגלל שיש קשתות משני הכיוונים, אפשר להגיע לכל הצמתים.
- סעיף 3 נכון- כדי שיהיה מעגל חייב להיכנס קשת לפחות לשני קודקודים.
- סעיף 4 - לא נכון. הדוגמה למשל ריבוע עם קשתות 2,5,3,4
- סעיף 5 – יתקבלו שתי תשובות נכון + נימוק (בעץ פורש מינימלי תמיד נבחרים הקשתות עם המשקל הנמוך יותר) / לא נכון + הוכחה נגדית (בגרף מכוון עם מעגל). תלמיד שענה אחרת, יינתן הניקוד עבור סעיף זה לארבעת הסעיפים האחרים שהוא ענה (15% לכל סעיף).

הטענה אינה מוגדרת היטב — עץ פורש מינימלי הוא מושג השייך לגרפים **לא מכוונים** בלבד. לכן:

- תלמיד שציין שהטענה **אינה מוגדרת** לגרף מכוון - יקבל ניקוד מלא.
- תלמיד שהתייחס לגרף **לא מכוון** (התעלם מהמילה "מכוון") וענה "נכון" עם נימוק שכאשר כל המשקלים שונים העץ הפורש המינימלי יחיד - יקבל ניקוד מלא.
- תלמיד שענה "לא נכון" עם דוגמה נגדית של גרף מכוון עם מעגל - יקבל ניקוד מלא.
- כל תשובה אחרת – יש להתעלם מהתשובה, ומשקל כל אחת מהטענות האחרות יהיה 15%
- סעיף 6 – לא נכון. בדוגמה שלפנינו יש רק מסלול אחד ולכן הוא תמיד ייבחר (יש הרבה דוגמאות שמפריכות את הטענה).



- סעיף 7 - לא נכון. יש לצייר מעגל.

סעיף ב – **40%** כל עץ – 10% (אם לא נכון אז לדוגמה יש 5%).

- G1 - לא נכון. אם מתחילים מצומת B ועוברים לצומת D, אז לצומת D יש שני בנים.
- G2 - לא נכון. לדוגמה, אם מתחילים מצומת C אז יהיו לו שני בנים.
- G3 – נכון.
- G4 – לא נכון. לדוגמה, אם מתחילים מצומת B אז יהיו לו שני בנים.
- G5 – לא נכון. לדוגמה, אם מתחילים מצומת B אז יהיו לו שלושה בנים.
- G6 – לא נכון. לדוגמה, אם עושים סריקה a->b->c->d אז ל c יש שני בנים.
- G7 –נכון.

שאלה 5

סעיף א - 20%

- כל תא – 5%

U	A1	A2	A3	B1	B2	B3
M	B1, B2	B1, B2, B3	B1, B2	A1, A2, A3	A1, A2, A3	A2

סעיף ב – 60%
כל שורה 10%

q	U	VISITED						PARENT					
		A1	A2	A3	B1	B2	B3	A1	A2	A3	B1	B2	B3
A3		לא	לא	כן	לא	לא	לא						
B1,B2	A3	לא	לא	כן	כן	כן	לא				A3	A3	
B2,A1	B1	כן	לא	כן	כן	כן	לא	B1			A3	A3	
A1,A2	B2	כן	כן	כן	כן	כן	לא	B1	B2		A3	A3	
A2	A1	כן	כן	כן	כן	כן	לא	B1	B2		A3	A3	
B3	A2	כן	כן	כן	כן	כן	כן	B1	B2		A3	A3	A2
	B3												

סעיף ג - 20%

B3, A2,B2,A3

שאלה 6

סעיף א – 30% (כל תת סעיף 10%)

- **סעיף 1:** דרגת כל קוד במעגל היא 3 כי היא מחוברת לשני קודקודי מעגל משני הצדדים ובנוסף היא מחוברת לקודקוד מרכז. קודקוד מרכז דרגתו n כי הוא מחובר לכל קודקודי המעגל.
- **סעיף 2:** יש n קשתות מעגל כי כל קודקודי המעגל מחוברים במעגל, ו n קשתות מרכז כי יש n קודקודים אליהם מחובר קודקוד מרכז.
- **סעיף 3:** הגרף לא דו צדדי. נימוק נניח בשלילה שזה גרף דו צדדי, ולכן שני קודקודי מעגל שביניהם קשת בקבוצות שונות. אך שני הקודקודים גם מחוברים לקודקוד מרכז.
הגדרה של גרף דו-צדדי אומרת שאפשר לחלק את כל קודקודי הגרף לשתי קבוצות נפרדות כך שאין אף קשת בין שני קודקודים שנמצאים באותה הקבוצה. קשתות יכולות לעבור רק מקבוצה אחת לשניה
גרף הוא דו-צדדי אם ורק אם אין בו מעגלים באורך אי-זוגי. הגרף מכיל תמיד משולשים או מעגלים אי-זוגיים חיצוניים

סעיף ב – 30%
קביעה – 15%, נימוק – 15% : עפ"ים יחיד. בעץ הפורש המינימלי יש רק קשתות מרכז שמשקלם 1.

סעיף ג (40%) :

- **סעיף ג(1)** – 20% : פתרון : כל תשובה שבה 2m<b.
נימוק : לכל מסלול קצר בין שני קודקודי מעגל העובר דרך קודקוד מרכז, יש שתי קשתות מרכזיות. המסלול הקצר ביותר שאינו עובר דרך קודקוד מרכז, הוא בין שני קודקודי מעגל המחוברים בקשת אחת של מעגל. לכן חייב להתקיים היחס הזה.
- **סעיף ג(2)** – 20%. הפתרון הוא 4 (פחות מ-5). נימוק : ניקח שני קודקודי מעגל שכנים שביניהם קשת מעגל אחת באורך 10. כדי שהמסלול המינימלי יעבור תמיד דרך קודקוד מרכז, צריך שסכום משקל של שני קשתות מרכז יהיו פחות ממשקל קשת מעגל.
הערה : כתבו משקל 5 - להוריד 5%.

מודלים חישוביים

שאלה 7

סעיף א – 30%

סעיף א(1) – 15% (כל שפה 5% :קביעה - 3%, נימוק 2%)

- L_1 רגולרית כי n קטן מ – 1000 ולכן יש מספר סופי של אפשרויות (ל k אין משמעות).
- L_2 לא רגולרית כי יש תלות בין מספר המופעים של- a למספר של- c, ו- n לא מוגבל.
- L_3 רגולרית כי n קטן מ - 2000.

סעיף א(2) – 15% (כל שפה 5% : קביעה 3%, נימוק 2%)

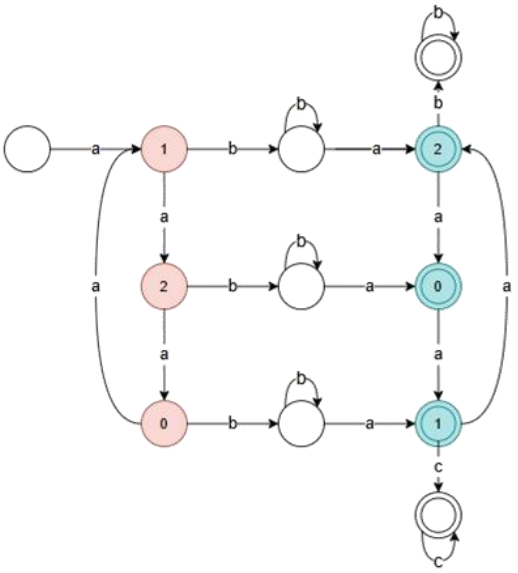
- L_4 רגולרית. החיתוך של שתי השפות הוא שפה ריקה כי הסדר של האותיות הפוך ($0 < n$ ולכן הרברס מחייב שיופיעו אותיות). שפה ריקה היא שפה רגולרית.
- L_5 לא רגולרית. L_2 לא רגולרית, והאיחוד דורש לקבל את השפה L_2 כמות שהיא (L_1 מתייחס רק למילים בהם n מתחת ל - 1000 ואינו מרחיב את L_2).
- L_6 רגולרית. השפות הרגולריות סגורות למשלים ולחיתוך (כל נימוק נכון יתקבל).

אם הגדירו את השפה המתקבלת וכתבו שפה שגוייה, לא לתת ניקוד (נימוק שגוי)

<u>סעיף ב – 70%</u>

- *קבלת כל המילים החוקיות*
- *דחיית מילים אסורות*

- טיפול ב n>0 – 5%
- טיפול ברצף ראשון של a (חילוק ב-3) – 15%
- טיפול ב m>0 – 5%
- טיפול ב k>0 – 5%
- טיפול ברצף שני של a (השלמה חילוק ב-3) – 25% סה"כ :
  - חילוק ב -3 10%
  - התאמת שאריות 15%
- טיפול במילה w בהתאם – 15% סה"כ :
  - ללא שארית (מצב מקבל - מילה ריקה)– 5%
  - שארית 1 (מצב מקבל + מעבר עם האות c למצב מקבל וקשת עצמית) – 5%
  - שארית 2 (מצב מקבל + מעבר עם האות b למצב חדש וקשת עצמית) – 5%



- לא התייחסו ל a גדול מ 0 – להוריד 5%
- על כל מצב מקבל חסר – להוריד 5%
- על כל מצב מקבל שגוי – להוריד 5%
- מעבר שגוי (למשל משארית 1 לשארית 0) – להוריד 10%
- לא סימנו את המצב ההתחלתי – להוריד 5%
- לא טיפלו ברצף - לא כתבו מעבר בודד ל a מרצף b – להוריד 10%
- כתב אוטומט לא דטרמיניסטי תקין – לתת לכל היותר 35%
- כתב אוטומט לא דטרמיניסטי שגוי – להוריד 10% בנוסף לכל שגיאה

שאלה 8

סעיף א – 10% (כל מילה – 2%)

- **abcc** שייכת לשפה. a אי זוגי ומתקיים התנאי: 1≤1+2-2
- **acccc** שייכת לשפה. a אי זוגי ומתקיים התנאי 1 ≤ 0+4-2
- **aabbc** שייכת לשפה. a זוגי, כל האותיות מופיעות, ו - 2=2/2+1
- **aabbbb** לא שייכת לשפה. a זוגי, והתו c לא מופיע (k>0)
- **aaabbc** לא שייכת לשפה. a אי זוגי, ולא מתקיים התנאי כי 3>2+1-2

סעיף ב – 10% (כל מילה 5%)

- המילה הקצרה ביותר עבור n זוגי היא aabbc
- המילים הקצרות ביותר בעבור n אי-זוגי: abbb, abbc,abcc,accc

סעיף ג - 80%

- שלושה מצבים מקבלים – 17% (כל מצב 6%)
- 21 מעברים– 63% (כל מעבר 3%)

1	b, A / לל״ש	b ,A / pop	b, A / Push AA	11	c, A / לל״ש c, S / לל״ש
2	b ,A / pop	b, A / לל״ש	b ,A / pop	12	c, A / לל״ש c, S / לל״ש
3	b, A / לל״ש	b ,A / pop	b, A / Push AA	13	b ,A / pop
4	c ,A / pop			14	c, A / pop
5	c, A / pop			15	c ,A / pop
6	c ,S / pop			16	b ,S / pop
7	c ,S / pop			17	c ,S / pop
8	b, A / לל״ש b, S / לל״ש			18	c ,S / pop
9	c, A / לל״ש c, S / לל״ש			19	b,⊥ / לל״ש
10	b, A / לל״ש b, S / לל״ש			20	c,⊥ / לל״ש
				21	c,⊥ / לל״ש

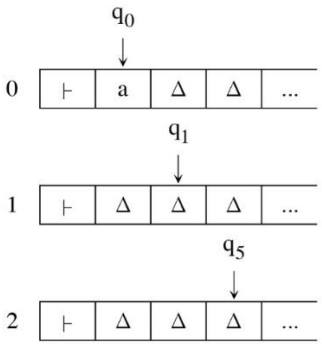
מצבים מקבלים : **q9, q10, q11**

- על שגיאה (או באות בראש המחסנית או בפעולה) להוריד 2% ולא יותר מסה״כ 3% למעבר.
- במצבים q8-q12, להוריד 1% על כל טעות אם שכחו לציין את לל״ש/S
- אם לא כתב לל״ש בכלל – להוריד 5% פעם אחת
- אם יש מעבר נוסף שלא גורר שגיאה – לא להוריד

שאלה 9

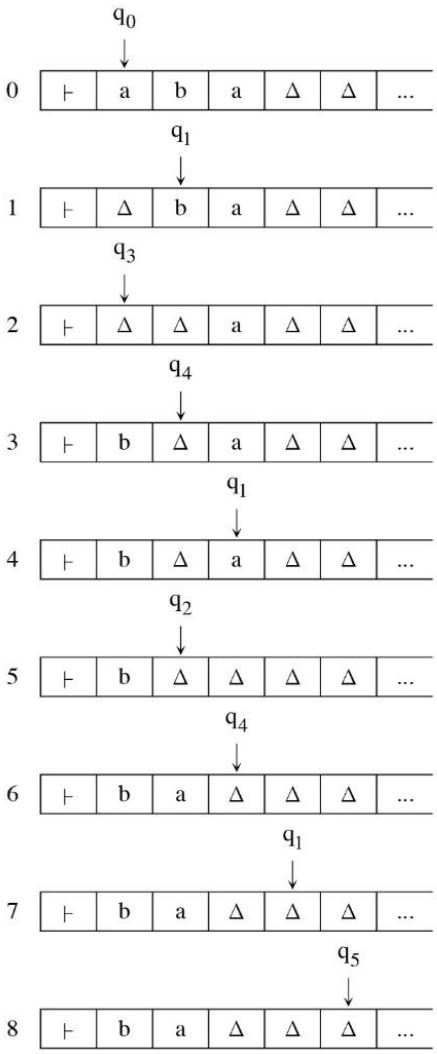
סעיף א – 80%

סעיף א (1) – 15% (3 שלבים - כל אחד – 5%)



- אם טעו להוריד 5% ולהמשיך לבדוק לשיטתם

סעיף א(2) – 45% (9 שלבים – כל אחד 5%)



- **שימו לב**, אם טעו להוריד 5% ולהמשיך לבדוק לשיטתם

סעיף א(3) - 20%

⊢	b	b	a	Δ

סעיף ב – 20% (כל סעיף 10%)

סעיף 1 – לא קיימת, כי המכונה מוחקת תו של a.
סעיף 2 – אם יש a המכונה מוחקת את ה-a השמאלי ביותר, ומבטלת את הרווח שנוצר.

- כתבו מוחקת תו של a ולא כתבו את השמאלי ביותר – להוריד 5%
- כתבו את כל התווים של a – להוריד 5%
- לא כתבו שמבטל את הרווח שנוצר– להוריד 3%

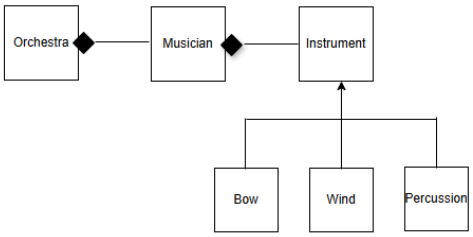
תכנות מונחה עצמים

שאלה 10

סעיף א – 28%

סעיף א(1) – 10%

- תרשים היררכיה – 10% (2% לכל חץ)



- כיוון חץ שגוי – להוריד 2% בכל טעות ולא יותר מ-6% סה"כ – מעויין באמצע = חץ שגוי
- סוג חץ שגוי – להוריד 2% בכל טעות ולא יותר מ – 6% סה"כ

סעיף א(2) – 18%

- כותרת ותכונות מחלקות –18% (כל מחלקה 3%)

```
public class Instrum{
    private/protected String name;
    private/protected String material;
}
public class Bow extends Instrum {
    private/protected int bowLength;
}
public class Wind extends Instrum {
    private/protected numberOfValves;
}
public class Perc extends Instrum {
    private/protected boolean hasSticks;
}
public class Musician{
    private/protected String name;
    private/protected Instrum instrum;
}
public class Orchestra{
    private/protected Musician[] arrM;
    private/protected int count;
}
```

- הוסיפו או הורידו תכונות –להוריד 2% לכל מחלקה
- לא הגדירו הרשאות גישה לתכונות או הגדירו public לתכונות – להוריד 1% לכל מחלקה
- לא הגדירו ירושה – להוריד 1% לכל מחלקה

סעיף ב – 22%

- לולאה עד count – 4%
- בדיקת סוג עצם – 5%
- המרה ובדיקה האם יש מקלות – 10% (המרה 5% בדיקה 5%)
- סכימה והחזרה – 3%

- לא בדקו סוג עצם – להוריד 10%
- לא עשו המרה – להוריד 5%
- גישה ללא get – להוריד פעם אחת 5%
- סריקה מעבר ל count-1 - להוריד 5%
- גלשו מהמערך – להוריד 10%

סעיף ג – 50%

- החזרה false אם המערך מלא – 5%
- הכנסת נגן כלי קשת (בהתחלה או במקום נכון אחר) – 10%
- הכנסת נגן כלי נשיפה (15%):
  - איתור מקום מתאים – 10%
  - הכנסה – 5%
- הכנסת נגן כלי הקשה (בסוף או במקום נכון אחר) – 10%
- עדכון המונה – 5%
- החזרה true – 5%

- הכניסו למקום מתאים אך לא הזיזו בכלל (דרסו את הערך שמופיע שם) – להוריד 5% בכל טעות
- הכניסו למקום מתאים והזיזו אך איבדו את הערכים (הכניסו לפני ששמרו) – להוריד 10%
- הכניסו תמיד לסוף או תחילת המערך (א בדקו בכלל סוג עצם)– להוריד 30%
- פספס מקום ב-1 – להוריד 5%
- השוו הפניות במקום טיפוסים או השוואה שגוייה אחרת– להוריד 10%
- השתמשו במערך עזר אך לא עדכנו את התכונה – להוריד 10%
- השתמש ב getClass() בג׳אווה או GetType() ב C# לא להוריד

שאלה 11

סעיף א – 25% (כל פעולה – 5%)

```
public Father (){
        st++;
        num=9;
}
public Father(int num){this.num=num; }
public Son (int num, char ch) {
        super(); //אפשר בלי
        this.num=num;
         this.ch=ch;
}
public Son(int num) {
        super (num);
         this.ch='Z';
}
public String toString() {return super.toString()+"ch="+this.ch; }
```

- כתבו את פעולת toString מהתחלה, ללא שימוש ב – super – להוריד 3%
- כתבו במחלקה Son את הפקודה ++st - להוריד 5%
- לא השתמשו ב super או לא השתמשו ב super הנכון – להוריד 5%
- לא אתחל תכונה או זימן super במקום הלא נכון – להוריד 2%
- טעות בכותרת הפעולה – להוריד 2%

סעיף ב – 40%

סעיף ב(1) – 30%

כל עץ – 5% (בעצים שלא ייתכן, הנימוק הוא 3%).

- עץ 1 : לא ייתכן. a2 מצביע לעצם מטיפוס B ו- a3 מצביע לעצם מטיפוס C ולשניהם ההפניה מטיפוס A, ולכן הם יורשים ממנו. מספיק נימוק אחד.
- עץ 2 : ייתכן.
- עץ 3 : לא ייתכן. C יורש מ A כאמור לעיל. בנוסף c2 מצביע על עצם מטיפוס D וההפניה מטיפוס C ולכן הוא יורש ממנו . מספיק נימוק אחד.
- עץ 4 : ייתכן.
- עץ 5 : לא ייתכן. B יורש מ – A כאמור לעיל.
- עץ 6 ייתכן.

סעיף ב(2) – 10% (כל שינוי – 5%)

a3 מטיפוס B והמצביע שלו מטיפוס D, לכן B יורש מ -D והעצים 2, 4 לא ייתכנו.

רק עץ 6.

סעיף ג – 35%

- שתי תכונות – 10% (כל תכונה – 5%)
- ארבע פעולות – 25% (פעולה בונה ריקה 7%, והשאר – 6%)

```
public class AAA{
        protected int num;
        protected static int c;
        public AAA(){…}
        public AAA(int num){…}
        public int gg()(int val) ///public virtual int GG(int val)
        public boolean mm(int num) //    public bool MM(int num)
}
```

- כתבו את num, או את c כתכונה פרטית – להוריד 3% על כל טעות
- כתבו את num, או את c כתכונה ציבורית – להוריד 1%
- לא כתבו ש- c הוא סטטי – להוריד 3%
- כתבו בכותרת הפעולה טיפוס מוחזר שגוי – להוריד 3%
- אם מימשו פעולות באופן שגוי, לא להוריד
- הגדיר פעולה מיותרת – להוריד 5%

שאלה 12

סעיף א – 30%

כל עצם – 5% (עצם – 1%, הפניה – 1%, num – 3%)

num	הפניה	עצם	
1	A	A	a
2	A	B	ab
2	A	C	ac
3	B	B	b
1	B	C	bc
1	C	C	c

סעיף ב – 70%

- כל הדפסה – 7%

1. -2
2. 0
3. 4
4. 22
5. 20
6. 8
7. 2
8. 21
9. 11
10. 4

- **שימו לב**, יש לבדוק את ההדפסות בהתאם לשיטתו (תכונות ועצם) בסעיף א.

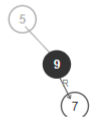
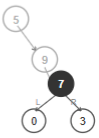
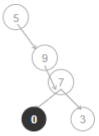
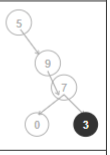
פתרונות נוספים

שאלה 3
סעיף א1

#	שלב	קריאה	left	right	תנאי / חישוב	מחזיר
1	ירידה	foo(5)	null	9	left==null → foo(right)+1 → קורא foo(9)	...ממתין
2	ירידה	foo(9)	null	7	left==null → foo(right)+1 → קורא foo(7)	...ממתין
3	ירידה	foo(7)	0	3	קורא → שני בנים foo(0)	...ממתין
4	ירידה	foo(0)	null	null	עלה: left==null AND right==null	0
5	חזרה	foo(7) ממשיך	0	3	קורא foo(3)	...ממתין
6	ירידה	foo(3)	null	null	עלה: left==null AND right==null	0
7	חזרה	foo(7) מסיים	0	3	foo(0)+foo(3) = 0+0	0
8	חזרה	foo(9) מסיים	null	7	foo(7)+1 = 0+1	1
9	**חזרה**	**foo(5) מסיים**	**null**	**9**	**foo(9)+1 = 1+1**	**2**

אפשרות נוספת :

סעיף א(1)

קריאה	left==null AND right==null	left==null	right==null	return
	False	True		foo(9)+1
	False	True		foo(7)+1
	False	False	False	foo(0)+foo(3)
	True			0
	True			0
חזרה: foo(7)	foo(0)+foo(3) = 0+0			0
חזרה: foo(9)	foo(7)+1 = 0+1			1
חזרה: foo(5)	**foo(9)+1 = 1+1**			**2**

סעיף א2

מחזיר	תנאי / חישוב	right	left	קריאה	שלב	#
...ממתין	foo(1) קורא → left==null → foo(right)+1	1	null	foo(5)	ירידה	1
...ממתין	foo(0) שני בנים → קורא	7	0	foo(1)	ירידה	2
...ממתין	foo(2) קורא → right==null → foo(left)−1	null	2	foo(0)	ירידה	3
...ממתין	foo(4) קורא → right==null → foo(left)−1	null	4	foo(2)	ירידה	4
0	עלה: left==null AND right==null	null	null	foo(4)	ירידה	5
−1	foo(4)−1 = 0−1	null	4	מסיים foo(2)	חזרה	6
−2	foo(2)−1 = −1−1	null	2	מסיים foo(0)	חזרה	7
...ממתין	foo(7) שני בנים → קורא	7	0	foo(7) ממשיך, קורא foo(1)	חזרה + ירידה	8
...ממתין	foo(3) קורא → left==null → foo(right)+1	3	null	foo(7)	ירידה	9
0	עלה: left==null AND right==null	null	null	foo(3)	ירידה	10
1	foo(3)+1 = 0+1	3	null	מסיים foo(7)	חזרה	11
−1	foo(0)+foo(7) = −2+1	7	0	מסיים foo(1)	חזרה	12
0	**foo(1)+1 = −1+1**	**1**	**null**	**מסיים foo(5)**	**חזרה**	**13**

אפשרות נוספת :

סעיף א(2)

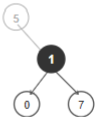
return	right==null	left==null	left==null AND right==null	קריאה
foo(1)+1		True	False	
foo(0)+foo(7)	False	False	False	
foo(2)−1	True	False	False	
foo(4)−1	True	False	False	
0			True	
1−	foo(4)−1 = 0−1			חזרה: foo(2)
2−	foo(2)−1 = −1−1			חזרה: foo(0)
foo(3)+1		True	False	
0			True	
1	foo(3)+1 = 0+1			חזרה: foo(7)
1−	foo(0)+foo(7) = −2+1			חזרה: foo(1)
0	foo(1)+1 = −1+1			חזרה: foo(5)

הפעולה מחזירה: 0

סעיף ב1

מחזיר	חישוב / תנאי	foo(tree)	tree==null?	קריאה	שלב	#
...ממתין	(של 5 goo(left קורא ← foo=0	0	לא	goo(5)	ירידה	1
true	tree==null	—	כן	[שמאל של 5] goo(null)	ירידה	2
...ממתין	right ממשיך לבדוק ← goo(null)=true			(של 5 goo(right ממשיך, קורא goo(5)	חזרה + ירידה	3
false	מיד false מחזיר ← 0 ≠ 1−=foo(1)	1−	לא	[ימין של 5] goo(1)	ירידה	4
false	**true && false**			**מסיים goo(5)**	**חזרה**	**5**

אפשרות נוספת

קריאה	tree==null	foo(tree)!=0	return
	False	False (foo(5)=0)	**goo(null) && goo(1)**
goo(null) [שמאל של 5]	True		**true**
	False	True (foo(1)=−1)	**false**
חזרה: goo(5)	**true && false**		**false**

הפעולה מחזירה: false