פתרון ממ"ן 12 – סמסטר 2020א

פתרון שאלה 1

א. נסמן ב-m את מספר ההופעות של x ברשימה A.

נתון: $m > n/2$

צ"ל: $m-1 > (n-2)/2$

זה נובע באופן מיידי מהנתון.

ב. הרצת האלגוריתם על המערך הנתון:

i	candidate	counter
1	3	1
2	3	2
3	3	1
4	3	0
5	3	1
6	3	0
7	4	1

הקריאה לשגרה Check-Majority-Element עם הערך candidate = 4 מחזירה תשובה שלילית; כלומר, אין איבר רוב במערך הקלט.

ג. האלגוריתם מחזיק במשתנה Candidate איבר שהוא **מועמד** להיות איבר רוב.

במשתנה Counter נשמר מספר ההופעות של Candidate.

בטענה מסעיף א' נעשה שימוש בשורה (2.3) באלגוריתם. לשורה זו מגיעים כאשר $counter \neq 0$ (כלומר, יש לנו מועמד לאיבר רוב), אך A[i] שונה מהמועמד הנוכחי. במקרה זה אנו מקטינים את counter ב-1 ולא עושים דבר עם A[i]. זה שקול בעצם למחיקה מהרשימה של A[i] ושל איבר אחד השווה לערכו של המועמד. הטענה מסעיף א' מבטיחה לנו שאם המועמד הוא אכן איבר רוב, הוא יישאר איבר רוב גם ברשימה החדשה. הבדיקה בשורה (4) נדרשת מפני שמועמד להיות איבר רוב לא חייב להיות איבר רוב (כמו בדוגמה הנתונה בשאלה), וכדי לדעת אם הוא איבר רוב צריך לספור את מספר ההופעות שלו ברשימה המקורית.

פתרון שאלה 2

א. אלגוריתם הפותר את הבעיה בזמן $O(n^2)$:

(1) הקצה מערך חדש B בגודל n והצב $B[1] \leftarrow A[1]$

(2) עבור i המקבל את הערכים 2 עד n בצע:

(2.1) $sum \leftarrow 0$

(2.2) עבור j המקבל את הערכים 1 עד i בצע:

(2.2.1) $sum \leftarrow sum + A[j]$

(2.3) $B[i] \leftarrow sum$

(3) החזר את B.

זמן הריצה: $1 + 2 + 3 + \ldots + n = O(n^2)$

ב. אלגוריתם הפותר את הבעיה בזמן $O(n)$:

(1) עבור i המקבל את הערכים 2 עד n בצע :

(1.1) $A[i] \leftarrow A[i-1] + A[i]$

(2) החזר את A.

הוכחת נכונות (בשיטת האינווריאנטות)

נשבץ באלגוריתם שתי נקודות ביקורת (לפני שורה 1 ולפני שורה 2) ושתי טענות ביניים.

<u>טענות הביניים</u> :

טענה 1 המתקיימת לפני שורה (1) :

לכל $1 \le j \le i-1$, $A[j]$ מכיל את סכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס j

טענה 2 המתקיימת לפני שורה (2) :

המערך A הוא מערך הפלט המבוקש

בפעם הראשונה שהאלגוריתם מגיע לנקודת הביקורת הראשונה ערכו של i הוא 2, ולכן טענה 1 היא בעצם :

$A[1]$ מכיל את סכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס 1

טענה זו היא כמובן נכונה.

<u>הוכחת המעברים בין טענות הביניים</u> :

טענה 1 ⇐ טענה 1 : לפני ביצוע האיטרציה שבה $i = k$ מתקיימת טענה 1 :

לכל $1 \le j \le k-1$, $A[j]$ מכיל את סכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס j

באיטרציה הבאה מתבצעת ההוראה $A[k] \leftarrow A[k-1] + A[k]$

לפי הנתון, $A[k-1]$ מכיל את סכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס k-1.

כאשר מוסיפים לו את הערך הנמצא בתא ה-k במערך המקורי מקבלים את סכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס k. כלומר, כעת גם האיבר $A[k]$ שווה לסכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס k. לכן כעת מתקיימת הטענה :

לכל $1 \le j \le k$, $A[j]$ מכיל את סכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס j

כלומר, טענה 1 נשארת נכונה עבור הערך החדש $i = k+1$.

טענה 1 ⇐ טענה 2 : בעת היציאה מהלולאה ערכו של i הוא n+1, ולכן טענה 1 היא בעצם :

לכל $1 \le j \le n$, $A[j]$ מכיל את סכום האיברים של המערך המקורי בין אינדקס 1 לאינדקס j

טענה זו שקולה לטענה שהמערך A הוא מערך הפלט המבוקש.

פתרון שאלה 3

א. **חיפוש טרנרי של** x **ברשימה** L

(1) אם L ריקה, אז חזור עם התוצאה "x לא נמצא ב-L".

(2) אחרת, בצע את הפעולות הבאות:

(2.1) חלק את L לשלושה חלקים שווים L_1, L_2, L_3

(2.2) הכנס ל- t_1 את האיבר האחרון ב- L_1, והכנס ל- t_2 את האיבר האחרון ב- L_2

(2.3) אם $x = t_1$, אז חזור עם התוצאה "x נמצא ב-L".

(2.4) אחרת, אם $x = t_2$, אז חזור עם התוצאה "x נמצא ב-L".

(2.5) אחרת, אם $x < t_1$, אז קרא ל- **חיפוש טרנרי של** x **ברשימה** L_1

(2.6) אחרת, אם $x > t_2$, אז קרא ל- **חיפוש טרנרי של** x **ברשימה** L_3

(2.7) אחרת, קרא ל- **חיפוש טרנרי של** x **ברשימה** L_2

(3) חזור.

ב. הוכחת נכונות: באינדוקציה על אורך הרשימה

<u>בסיס האינדוקציה</u>: עבור רשימה ריקה האלגוריתם בוודאי מחזיר את התשובה הנכונה.

<u>צעד האינדוקציה</u>: תהי L רשימה באורך n. נניח שהאלגוריתם נכון לכל רשימה באורך $k < n$.

אם האלגוריתם מוצא את x בשורה (2.3) או (2.4), אז הוא בוודאי מחזיר את התשובה הנכונה.

אם מתקיים התנאי בשורה (2.5), אז x יכול להימצא רק בתת-רשימה L_1 (כי הרשימה L ממוינת). במקרה זה האלגוריתם מבצע קריאה רקורסיבית עם L_1, ומהנחת האינדוקציה נובע שתוחזר התשובה הנכונה. באופן דומה, אם מתקיים התנאי בשורה (2.6), אז x יכול להימצא רק בתת-רשימה L_3. במקרה זה האלגוריתם מבצע קריאה רקורסיבית עם L_3, ומהנחת האינדוקציה נובע שתוחזר התשובה הנכונה. אם התנאים בשורה (2.5) ו-(2.6) אינם מתקיימים, אז x יכול להימצא רק בתת-רשימה L_2. במקרה זה האלגוריתם מבצע קריאה רקורסיבית עם L_2, ומהנחת האינדוקציה נובע שתוחזר התשובה הנכונה.

זמן הריצה: בכל קריאה רקורסיבית אורך הרשימה קטן פי 3, ולכן זמן הריצה הוא $O(\log_3 n)$.

פתרון שאלה 4

א. אלגוריתם יעיל לפתרון הבעיה: ראשית ממיינים את n הקטעים במיון-מיזוג לפי נקודות ההתחלה שלהם.
בשלב השני עוברים על הרשימה הממוינת של הקטעים, ובודקים אם קיים $1 \leq i < n$ כך שמתקיים $a_{i+1} \leq b_i$.
אם קיים i כזה, אז האלגוריתם מחזיר תשובה חיובית; אחרת, הוא מחזיר תשובה שלילית.

ב. הסבר: אם יש חפיפה בין שני קטעים, זאת אומרת שהקטע "המאוחר יותר" מבין שניהם מתחיל לפני שהקטע הקודם הסתיים. זה מה שבודקים בשלב השני של האלגוריתם.

זמן הריצה: $O(n\log n) + O(n) = O(n\log n)$

פתרון שאלה 5

כדור 1 – מבצעים קפיצות של $n^{2/3}$ קומות. במקרה הגרוע יתבצעו $n^{1/3}$ קפיצות.

כדור 2 – מבצעים קפיצות של $n^{1/3}$ קומות (בקטע המורכב מ- $n^{2/3}$ קומות). במקרה הגרוע יתבצעו $n^{1/3}$ קפיצות.

כדור 3 – מבצעים קפיצות של 1 קומה (בקטע המורכב מ- $n^{1/3}$ קומות). במקרה הגרוע יתבצעו $n^{1/3}$ קפיצות.

סיבוכיות הזמן: $3 \times n^{1/3} = O(n^{1/3})$