



תאריך: מרץ-2026

צבי מלמד – דף תירגול מדעי המחשב כיתה יא-1 תשפ"ו



מבוא למדעי המחשב – כיתה יא-1 שאלות לתרגול מבחינות הבגרות

הנושא: תורים

בחינה : 2021 מועד א'

שאלה מספר: 7

השאלה עוסקת ב-:

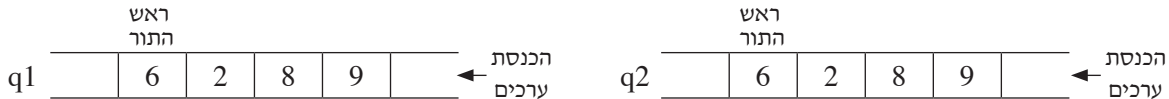
- האם שני תורים הם זהים
- האם שני תורים הם "דומים" – העברה ROTATE של איברים

7. בשאלה זו תוכל להשתמש בפעולה החיצונית שלפניך בלי לממש אותה.

תיאור הפעולה	כותרת הפעולה
הפעולה מחזירה את מספר האיברים בתור q בלי לשנות את התור.	בשפת Java — <code>public static int size (Queue<Integer> q)</code> בשפת C# — <code>public static int Size (Queue<int> q)</code>

א. שני תורים, q_1 ו- q_2 , יהיו "תורים זהים" אם מספר האיברים בשני התורים זהה, ובשני התורים מופיעים בדיוק אותם ערכים ובאותו הסדר.

דוגמה לשני תורים זהים:

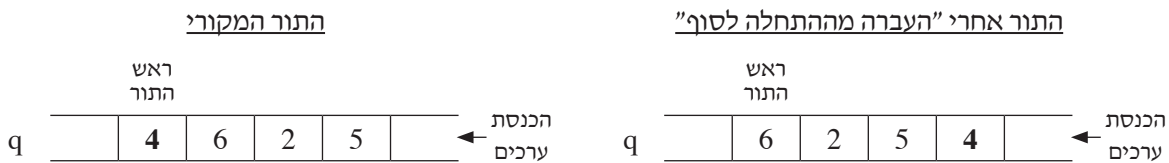


כתוב פעולה חיצונית ששמה `isIdentical` בשפת `Java` או `IsIdentical` בשפת `C#` המקבלת שני תורים מטיפוס שלם q_1 ו- q_2 , ומחזירה `true` אם התורים זהים, אחרת היא מחזירה `false`.

הערה: עם סיום הפעולה, חובה לשמור על מבנה התורים המקורי שהתקבל.

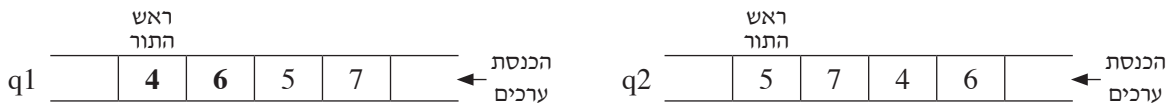
ב. "העברה מההתחלה לסוף" היא העברת מספר מראש התור לסופו.

דוגמה:



כתוב פעולה חיצונית ששמה `isSimilar` בשפת `Java` או `IsSimilar` בשפת `C#` המקבלת שני תורים מטיפוס שלם, q_1 ו- q_2 . הפעולה מחזירה `true` אם התורים q_1 ו- q_2 זהים – בין שהם זהים כמו שהם ובין שהם יהיו זהים לאחר שנבצע ב- q_1 "העברה מההתחלה לסוף", פעם אחת או יותר. אחרת הפעולה מחזירה `false`.

דוגמה: הפעולה תחזיר `true` בעבור שני התורים q_1 ו- q_2 שלפניך:



זאת מכיוון שלאחר שבתור q_1 נבצע פעמיים "העברה מההתחלה לסוף", הוא יהיה זהה לתור q_2 , וייראה כך:



הערות: — חובה להשתמש בפעולה שכתבת בסעיף א.

— אין צורך לשמור על מבנה התורים המקורי שהתקבל.

תאריך: מרץ-2026

מבוא למדעי המחשב – כיתה יא-1 שאלות לתרגול מבחינות הבגרות

הנושא: תורים

בחינה : 2021 מועד ב'

שאלה מספר: 6

השאלה עוסקת ב-:

- CovidTest - מחלקה בשם הזה
- מעבר על תור של אובייקטים
- עבודה נכונה עם פעולות גישה למחלקה

6. נתונה המחלקה **CovidTest**, המייצגת אדם שנבדק בדיקת קורונה, ולה 4 תכונות:

- name – שם הנבדק מטיפוס מחרוזת
 - id – מספר זהות מטיפוס מחרוזת
 - cityCode – קוד של עיר המגורים, מטיפוס שלם (לדוגמה: 1030 בעבור אשדוד, 23 בעבור עכו)
 - sick – משתנה מטיפוס בוליאני, המקבל true אם הנבדק חולה, אחרת הוא מקבל false
- הנח שיש פעולות get ו-set בשפת Java ופעולות Get ו-Set בשפת C# בעבור תכונות המחלקה.
- כתוב פעולה חיצונית mostSick בשפת Java או MostSick בשפת C# המקבלת תור – q שאינו ריק מטיפוס **CovidTest**. הפעולה תחזיר את הקוד של העיר שבה כמות החולים היא הגדולה ביותר.

הערות: – מיקום הנבדקים בתור אינו לפי סדר כלשהו.

– כל נבדק מופיע רק פעם אחת בתור.

– הקוד של העיר אינו קשור לגודל התור (לדוגמה: ייתכן שמספר האיברים בתור הוא 1,000,

וקיים קוד עיר שמספרו 5000).

– אין צורך לשמור על התור.

הנח שיש רק עיר אחת שבה כמות החולים היא הגדולה ביותר.

תאריך: מרץ-2026

מבוא למדעי המחשב – כיתה יא-1 שאלות לתרגול מבחינות הבגרות

הנושא: תורים

בחינה : 2023

שאלה מספר: 4

השאלה עוסקת ב-:

- "מעבר מקונן" על תור
- האם קיימים שני איברים בתור שסכומם הוא X

פרק שני (50 נקודות)

שימו לב: בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

ענו על שתיים מן השאלות 4-6 (לכל שאלה – 25 נקודות).

4. בשאלה זו נוספה למחלקה **Queue** הפעולה `size/Size` שלפניכם. אפשר להשתמש בפעולה בלי לממש אותה.

תיאור הפעולה	כותרת הפעולה
הפעולה מחזירה את מספר האיברים בתור.	Java – <code>public int size()</code> C# – <code>public int Size()</code>

ממשו את הפעולה החיצונית שלפניכם:

Java – `public static boolean twoSum (Queue<Integer> q, int x)`

C# – `public static bool TwoSum (Queue<int> q, int x)`

הפעולה מחזירה `true` אם בתור `q` שהתקבל יש שני מספרים שסכומם שווה לערך הפרמטר `x`. אחרת הפעולה מחזירה `false`.
דוגמה: עבור התור `q` שלפניכם ו- `x = 10` הפעולה תחזיר `true`, כי יש בתור שני מספרים (9, 1) שסכומם שווה ל-10.

	ראש התור							
q	5	4	1	4	3	15	9	

הערות:

- הניחו שבתור `q` יש שני איברים לפחות.
- אין צורך לשמור על התור `q`.
- אין להשתמש בשאלה זו במערך וברשימה מקושרת.

תאריך: מרץ-2026

מבוא למדעי המחשב – כיתה יא-1 שאלות לתרגול מבחינות הבגרות

הנושא: תורים

בחינה : 2024

שאלה מספר: 4

השאלה עוסקת ב-:

- "מעבר על תור
- האם מספר במקום מסוים בתור מקיים שהוא שווה לקודמו ולזה שאחריו

יש לענות על שלוש שאלות סך הכול מן הפרקים השני והשלישי (לכל שאלה – 25 נקודות).

פרק שני

שימו לב: בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

4.

"איבר קסם" הוא איבר בתור של מספרים שערכו שווה לסכום הערכים של האיבר שלפניו והאיבר שאחריו.

הערה: המספר הראשון בתור והמספר האחרון בתור אינם "איברי קסם".

א. כתבו פעולה ששמה isMagic בשפת Java או IsMagic בשפת C#, המקבלת תור – q מטיפוס שלם,

ומספר שלם – m הגדול מ-0 וקטן או שווה לגודל התור.

הפעולה תחזיר true אם האיבר במקום ה- m בתור הוא "איבר קסם", אחרת היא תחזיר false.

הערות: – בסיום הפעולה חובה לשמור על מבנה התור כפי שהתקבל.

– אין להשתמש בסעיף זה במערך או ברשימה מקושרת. פתרון הכולל שימוש בהם לא יזוכה בנקודות.

לדוגמה: עבור התור שלפניכם:

ראש התור						סוף התור
1	2	3	4	5	6	7
5	11	6	9	3	6	3

עבור $m = 1$ הפעולה תחזיר false (המספר הראשון בתור אינו "איבר קסם")

עבור $m = 2$ הפעולה תחזיר true ($5 + 6 = 11$)

עבור $m = 3$ הפעולה תחזיר false ($11 + 9 \neq 6$)

ב.

כתבו פעולה ששמה nMagic בשפת Java או NMagic בשפת C# המקבלת תור מטיפוס שלם – q, ומספר שלם – n

הגדול מ-0 וקטן או שווה לגודל התור.

הפעולה תחזיר true אם כל האיברים הנמצאים במקומות שהם כפולה של n (המקום ה- n בתור, המקום ה- $2n$ בתור

וכן הלאה בדילוגים של n מקומות) הם "איברי קסם". אחרת הפעולה תחזיר false.

אפשר להשתמש בפעולה שכתבתם בסעיף א.

הערות: – בפעולה זו אין צורך לשמור על התור שהתקבל.

– אין להשתמש בסעיף זה במערך או ברשימה מקושרת. פתרון הכולל שימוש בהם לא יזוכה בנקודות.

דוגמאות: עבור התור שבדוגמה שלעיל:

עבור $n = 2$ הפעולה תחזיר true מכיוון שכל האיברים הנמצאים במקומות שהם כפולה של 2 (2, 4, 6) הם

"איברי קסם".

עבור $n = 4$ הפעולה תחזיר true מכיוון שהאיבר במקום ה-4 הוא "איבר קסם" (אין בתור איברים נוספים במקומות

שהם כפולה של 4).

עבור $n = 3$ הפעולה תחזיר false מכיוון שהאיבר במקום ה-3 אינו "איבר קסם".

תאריך: מרץ-2026

מבוא למדעי המחשב – כיתה יא-1 שאלות לתרגול מבחינות הבגרות

הנושא: תורים

בחינה : 2024

שאלה מספר: 5

(וגם: שאלה #2 בטופס 899281 באותה שנה)

השאלה עוסקת ב-:

- תור עדיפויות של אובייקטים ממחלקה נתונה (מופיעה בשאלה)
- הכנסת איבר חדש לתור עדיפויות
- שינוי עדיפות של איבר קיים בתור עדיפויות
- בנוסף לעבודה עם תורים זה מצריך עבודה נכונה עם פעולות של המחלקה הנתונה.

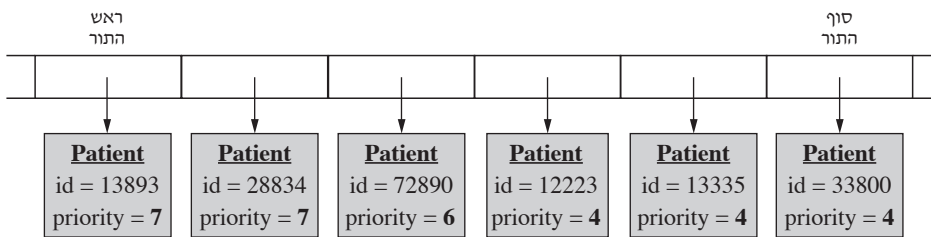
5. נתונה המחלקה **Patient** – חולה בחדר מיון, ולה שתי תכונות:

- id – מספר הזהות של החולה, מטיפוס שלם
 - priority – רמת הדחיפות של הטיפול בחולה. רמת הדחיפות מיוצגת במספר מטיפוס שלם בין 1 ל-10. ככל שהמספר גבוה יותר, רמת הדחיפות גבוהה יותר.
- הניחו שלתכונות המחלקה יש פעולות `get/Set` ו- `set/Set`.
- סדר הטיפול בחולים בחדר המיון מתנהל באופן שלהלן:
 ככל שרמת הדחיפות של הטיפול גבוהה יותר, החולה מטופל מוקדם יותר. כאשר יש יותר מחולה אחד באותה רמת דחיפות, החולה שהגיע קודם לחדר מיון מטופל מוקדם יותר.

כדי לשמור על סדר הטיפול נבנתה המחלקה **PriorQueue** – תור עדיפויות, ולה תכונה אחת:

- q – הפניה לתור, מטיפוס **Patient**

דוגמה לתור q:



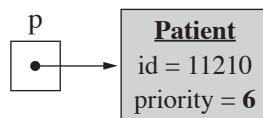
הניחו שהחולים שרמת הדחיפות שלהם זהה מוצגים בדוגמה לפי סדר הגעתם לחדר המיון.

א. ממשו את הפעולה שלפניכם השייכת לממשק המחלקה **PriorQueue**:

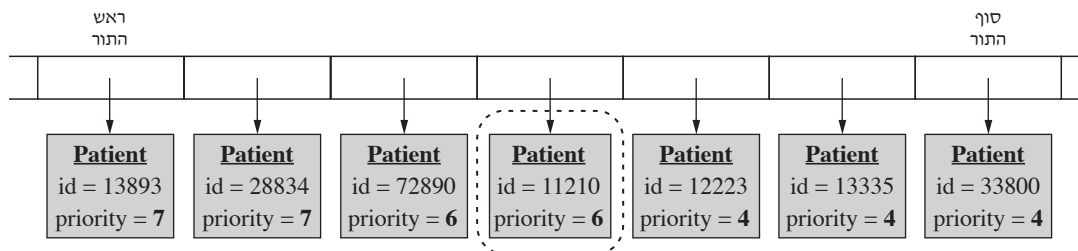
Java – `public void priorityInsert (Patient p)`

C# – `public void PriorityInsert (Patient p)`

הפעולה מקבלת חולה חדש – **p** ומכניסה אותו לתור **q** בהתאם לכללים של חדר המיון הכתובים לעיל.
לדוגמה: עבור התור המוצג לעיל והעצם שלפניכם:



התור ייראה כך לאחר ההכנסה:



(שימו לב: המשך השאלה בעמוד הבא.)

ב. מדי פעם רמת הדחיפות של חולה מסוים משתנה במהלך שהותו בחדר המיון.

ממשו את הפעולה שלפניכם השייכת לממשק המחלקה PriorityQueue :

Java – public void update (int id, int pri)

C# – public void Update (int id, int pri)

הפעולה מקבלת מספר זהות של חולה הנמצא בתור – id , ומספר – pri המייצג את רמת הדחיפות המעודכנת שלו. הפעולה תעדכן את התכונה priority של החולה ותמקם אותו בתור לטיפול במקום המתאים לו (בהתאם לרמת הדחיפות המעודכנת – pri). אם בתור כבר יש חולים אחרים באותה רמת דחיפות – pri , החולה הנוכחי – id יוצב אחריהם כאילו הגיע אחריהם לחדר המיון.