

מבני נתונים 234218 אביב 2023



גיליון יבש 3 – מעודכן לתאריך 11.06.2023

עידו גליל, idogalil@cs.technion.ac.il

מתרגל ממונה על התרגיל:

יום חמישי, 06.07.2023 בשעה 23:59

תאריך ושעת הגשה:

בזוגות.

אופן ההגשה:

הנחיות לפתרון:

- שאלות לגבי תרגיל הבית נא לשאול ב-Piazza של הקורס, מידע נוסף נמצא באתר.
- בקשות להגשה מאוחרת יש לשלוח למתרגלת האחראית של הקורס – סאלי.
- הגישו פתרון מוקלד. הגשה בכתב יד היא באישור המתרגל האחראי על התרגיל בלבד.
- כתבו את פתרונותיכם באופן מסודר ומובנה. בשאלות שעוסקות במבני נתונים יש להתחיל מפתח שמתאר את התשובה בקצרה (2-3 שורות) ולאחר מכן להציג את הפתרון המלא.
- הגשת התרגיל היא אלקטרונית בלבד, באתר הקורס, בקובץ PDF בלבד.
- **שימו לב:** הקפידו לצרף את כל השאלות והסעיפים לפי הסדר! **אי עמידה בכלל זה תגרור הורדה של ציון.**
- **שימו לב:** הקפידו לכתוב את פתרונותיכם באופן מסודר ומובנה. התחילו מפתח המתאר את תשובתכם בקצרה (עד 3 שורות) ולאחריו כתבו פירוט מלא של הפתרון. **אי עמידה בכלל זה תגרור הורדה של ציון.**
- אין צורך לפרט דברים שנלמדו בהרצאות או בתרגולים. מספיק לצטט או להפנות לחומר הלימוד. עם זאת, יש להוכיח כל טענה שלא נלמדה בהרצאה או בתרגול.
- יש לנתח סיבוכיות זמן ומקום של כל אלגוריתם ומבנה נתונים, אלא אם כן צוין אחרת.
- סיבוכיות זמן ומקום הן במקרה הגרוע, אלא אם כן צוין אחרת.

תמיכה ושאלות על הגיליון:

- **שימו לב:** בפורום הפיאצה ינוהל FAQ ויועלו תיקונים, במידת הצורך, כהודעות נעוצות (Pinned Notes). תיקונים אלו מחייבים.



שאלה 1 (15 נקודות)

הוכח \ הפוך: קיים מימוש לערימת מינימום המממש את פעולות הערימה בדרישות הסיבוכיות הבאות:

MakeHeap – יצירת המבנה. סיבוכיות זמן: $O(n)$ במקרה הגרוע.

Insert(x) – הכנסת איבר x למבנה. סיבוכיות זמן: $O(\log n)$ במקרה הגרוע.

FindMin – החזרת האיבר המינימלי במבנה. סיבוכיות זמן: $O(1)$ במקרה הגרוע.

DelMin – הוצאת האיבר המינימלי מהמבנה. סיבוכיות זמן: $O(\log n)$ במקרה הגרוע ו- $O(\log \log n)$ משוערך.



שאלה 2 (25 נקודות)

בשאלה זו נרצה לתכנן מבני נתונים אשר מאפשרים שמירה של הגרסאות השונות של המבנה לאורך ההיסטוריה. לצורך הפשטות כל הוספת או הסרת איבר מגדירה נקודת זמן בהיסטוריה, כלומר נקודת הזמן t היא מיד לאחר ביצוע t פעולות הכנסה והוצאה מהמבנה.

הציעו מבנה נתונים התומך בפעולות הבאות:

$Init()$ הפעולה מאתחלת מבנה ריק.

סיבוכיות זמן: $O(1)$.

$Insert(x)$

הפעולה מכניסה איבר למבנה.

סיבוכיות זמן: $O(\log n_x)$ בממוצע, משוערך, כאשר n_x הוא מספר הפעולות אשר בוצעו על האיבר x לאורך ההיסטוריה.

$Remove(x)$

הפעולה מסירה איבר מהמבנה.

סיבוכיות זמן: $O(\log n_x)$ בממוצע, כאשר n_x הוא מספר הפעולות אשר בוצעו על האיבר x לאורך ההיסטוריה.

$Find(x, t)$

הפעולה בודקת האם האיבר x היה במבנה בנקודת הזמן t .

סיבוכיות זמן: $O(\log n_x)$ בממוצע, כאשר n_x הוא מספר הפעולות אשר בוצעו על האיבר x לאורך ההיסטוריה.

סיבוכיות מקום: $O(k)$, כאשר k הוא מספר פעולות ההכנסה וההוצאה שבוצעו על כל האיברים במבנה. דוגמה:

```
Init()
Insert(15) //t = 1
Insert(30) //t = 2
Insert(20) //t = 3
Remove(30) //t = 4
Insert(30) //t = 5
Find(30,1) // False
Find(30,2) // True
Find(30,3) // True
Find(30,4) // False
Find(30,5) // True
```



שאלה 3 (25 נקודות):

בחברת מבני מבוכים © עוסקים בתכנון מבוכים ללקוחות. בחברה זו בונים מבוכים על ידי יצירת מטריצה של $n \times n$ תאים, ואז מסירים באופן הדרגתי קירות בין תאים במבוך. אחד המוצרים אותם מציעה החברה הוא "מבוך חסר מעגלים", במבוך מסוג זה בין כל שני תאים עובר רק מסלול אחד. מוצר נוסף אותו מציעה החברה הוא "מבוך קשיר", במבוך קשיר קיים מסלול בין כל זוג תאים. הציעו מימוש למבנה נתונים התומך בפעולות הבאות:

Init(n)

אתחל מבוך בגודל $n \times n$, בעת האתחול כל הקירות בין שני תאים סמוכים קיימים.

סיבוכיות זמן: $O(1)$.

RemoveBarrier((x₁, y₁), (x₂, y₂))

חסר את הקיר בין התאים בעלי הקואורדינטות (x_1, y_1) , (x_2, y_2) . ניתן להניח כי הקלט חוקי ושני התאים הם תאים סמוכים.

סיבוכיות זמן: $O(1)$.

IsAcyclic()

הפעולה מחזירה האם המבוך הוא מבוך חסר מעגלים.

סיבוכיות זמן: $O(n^2 \log^*(n^2))$ במקרה הגרוע.

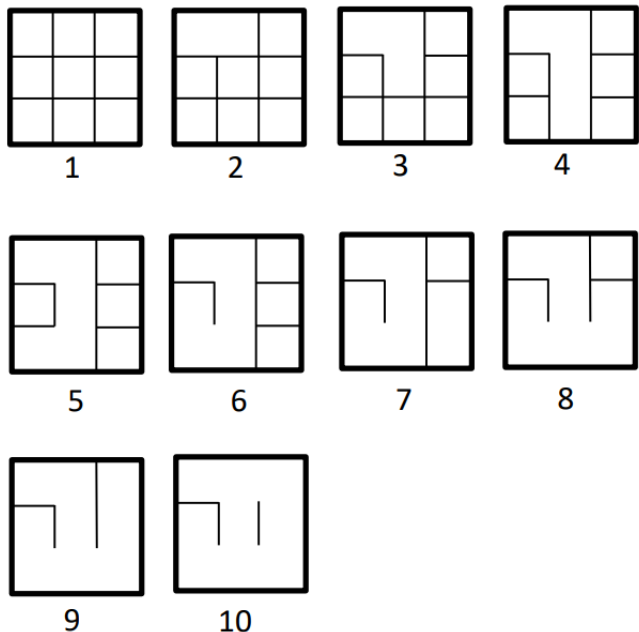
IsConnected()

הפעולה מחזירה האם המבוך הוא מבוך קשיר.

סיבוכיות זמן: $O(n^2 \log^*(n^2))$ במקרה הגרוע.

דוגמה (לצרכי קיצור נסמן את הפונקציה RemoveBarrier ב-RB):

- 1) Init(3)
- 2) RB((0,0), (0,1))//Acyclic, Not connected
- 3) RB((0,1), (1,1))//Acyclic, Not connected
- 4) RB((1,1), (2,1))//Acyclic, Not connected
- 5) RB((2,0), (2,1))//Acyclic, Not connected
- 6) RB((1,0), (2,0))//Acyclic, Not connected
- 7) RB((1,2), (2,2))//Acyclic, Not connected
- 8) RB((2,1), (2,2))//Acyclic, Not connected
- 9) RB((0,2), (1,2))//Acyclic, Connected
- 10) RB((0,1), (0,2))//Cyclic, Connected





שאלה 4 (35 נקודות):

במשחק שבץ-נא (סקראבל) שחקנים נדרשים לשבץ אותיות על לוח משחק כך שיצרו מילים חוקיות. במהלך המשחק שחקן יכול להרחיב מילה ששחקן אחר שם על מנת לקבל מילה אחרת, לכן, כאשר שחקן משבץ מילה הוא לא מעוניין שיהיה אפשר להרחיב אותה למילה אחרת כדי למנוע מהשחקן השני לקבל ניקוד בקלות.

השחקנים במשחק מעוניינים ליצור כלי אשר יזהה מילים שהן תחיליות של מילים אחרות.

סעיף א' (20 נקודות)

הציעו מבנה נתונים התומך בפעולות הבאות:

$Init()$	אתחל מבנה נתונים ריק.
	<u>סיבוכיות זמן:</u> $O(1)$
$Insert(s)$	הוסף מחרוזת s , למבנה הנתונים.
	<u>סיבוכיות זמן:</u> $O(s)$
$Remove(s)$	הסר את המחרוזת s ממבנה הנתונים.
	<u>סיבוכיות זמן:</u> $O(s)$
$PrefixExists()$	הפעולה מחזירה האם קיים זוג מחרוזות s_1, s_2 כך ש- s_1 היא תחילית של s_2 .
	<u>סיבוכיות זמן:</u> $O(1)$

סעיף ב' (15 נקודות)

הראה כיצד לשנות את הפעולה $PrefixExists$ באופן המתואר מטה. במידה ונדרשים שינויים בפעולות מהסעיף הקודם הקפידו להסבירם.

$PrefixExists(s)$	הפעולה מחזירה האם קיים זוג מחרוזות (שונות זו מזו) s_1, s_2 כך ש- s היא תחילית של s_1 ו- s_1 היא תחילית של s_2 .
	<u>סיבוכיות זמן:</u> $O(s)$