

## מבני נתונים – פרויקט מספר 2 – ערמת פיבונאצ'י

### הקדמה

בתרגיל זה שני חלקים:

1. חלק מעשי: מימוש של ערמת פיבונאצ'י. עמודים 1-2 במסמך מתארים חלק זה.
2. חלק ניסויי-תאורטי: בהתבסס על המימוש מהחלק המעשי, נבצע מספר ניסויים עם ניתוח תאורטי נלווה. עמוד 3 מתאר חלק זה.

שימו לב: בסוף המסמך (עמוד 3) ישנן הוראות הגשה – הקפידו לפעול לפיהן. **תאריך הגשה: 29/06/2025**

### חלק מעשי

#### דרישות

בתרגיל זה יש לממש ערמת פיבונאצ'י, לפי ההגדרות שניתנו בכיתה. לכל איבר במבנה יש מפתח (key) שהוא מספר טבעי ומידע (info) שהוא מחרוזת. המפתחות לא בהכרח ייחודיים, וכלל הערמה מתייחס כרגיל אך ורק למפתחות. המימוש יהיה בשפת **Java 21** וצריך להיות מבוסס על קובץ השלד המופיע באתר הקורס.

בכיתה למדנו שבערימות פיבונאצ'י כאשר צומת מאבד שני בנים אז אנו חותכים את הצומת (תהליך cascading cuts). במטלה הזו אנו נשנה את המימוש כך שנחתוך צומת רק אם הוא איבד  $c$  בנים (עבור  $c \geq 2$  כלשהו). המימוש שלמדנו בכיתה הוא עבור  $c=2$ . שימו לב שהצומת שעליו עושים decrease-key תמיד נתלש (אם הוא מפר את תכונת הערימה) בלי קשר לכמות הבנים שאיבד.

הפעולות שיש לממש הן:

פְּעוּלָה	תִּיאוֹר
<b>FibonacciHeap(c)</b>	הבנאי של המחלקה יקבל כפרמטר את הערך $c \geq 2$ שתואר לעיל.
<b>insert(k, info)</b>	הכנסת איבר בעל מפתח k לערמה עם מידע info. הפונקציה מחזירה מצביע לצומת בערימה שנוצר עבורו.
<b>findMin()</b>	הפונקציה מחזירה את איבר הערמה בעל המפתח המינימלי.
<b>deleteMin()</b>	מחיקת האיבר המינימלי מהערמה (אין צורך להחזיר). יש להחזיר את מספר הלינקים שבוצעו (במסגרת consolidating / successive linking שלמדנו).
<b>decreaseKey(x, d)</b>	הפונקציה מקבלת מצביע לאיבר הערמה x וטבעי d. היא מפחיתה את המפתח של x ב-d ומתקנת את הערמה כפי שלמדנו בכיתה. יש להחזיר את מספר הפעמים שבהם צומת התנתק מאביו (הוצא מרשימת הבנים שלו והוסף לרשימת השורשים) שבוצעו (במסגרת cascading cuts, כולל החיתוך הראשוני של הצומת x, אם קרה).
<b>delete(x)</b>	הפונקציה מקבלת מצביע לאיבר הערמה x ומוחקת אותו מהמבנה. יש להחזיר את מספר הלינקים שבוצעו. ממשו פעולה זו באמצעות קריאה לפעולות אחרות מהרשימה הזו.
<b>totalLinks()</b>	הפונקציה מחזירה את סך החיבורים (links) המצטבר שבוצעו (חיבור שני עצים מאותה הדרגה) מרגע יצירת הערימה.
<b>totalCuts()</b>	הפונקציה מחזירה את סך החיתוכים (ניתוק קשת) המצטבר שבוצעו מרגע יצירת הערימה, כלומר את מספר הפעמים שבהם צומת התנתק מאביו (הוצא מרשימת הבנים שלו) שבוצעו. שימו לב שניתוק של צומת מאביו יכול לקרות כאשר קוראים decreaseKey אבל גם כאשר קוראים delete.
<b>meld(heap2)</b>	הפונקציה ממזגת את הערמה עם ערמה נוספת heap2. לאחר הקריאה לפעולה הערמה heap2 אינה שמישה.
<b>size()</b>	הפונקציה מחזירה את מספר האיברים בערמה.
<b>numTrees()</b>	הפונקציה מחזירה את מספר העצים בערמה.

שמישה ? ←

לצורך מימוש פעולות אלו, ניעזר במחלקה **HeapNode** המופיעה בקובץ. המחלקה HeapNode המייצגת צומת בערמה מכילה את השדות הבאים:

- key – המפתח ששמור בצומת זה.
- info – המידע ששמור בצומת זה.
- child – בן כלשהוא של צומת זה.
- next – האח הבא של צומת זה.
- prev – האח הקודם של צומת זה.
- parent – ההורה של צומת זה בערמה.
- rank – דרגת הצומת (מספר הבנים).

#### הערות חשובות:

1. המימוש יבוצע על ידי מילוי קובץ השלד. מותר להחליף את תוכן הפונקציות הקיימות ולהוסיף פונקציות ושדות חדשים. אסור לשנות את חתימות הפונקציות הקיימות ואת שמות השדות הקיימים כדי לא לפגוע בטסטר. על כל הפונקציות/מחלקות להופיע בקובץ יחיד.
2. אין להשתמש באף מימוש ספרייה של מבנה נתונים.
3. עליכם לממש את כל הפעולות בסיבוכיות המיטבית.

#### סיבוכיות

מלאו את הטבלה הבאה. מספר האיברים בערימה הוא  $n$ . התייחסו ל $c$  כקבוע בניתוח.

זמן ריצה לשיעורין	זמן ריצה במקרה הגרוע	פעולה
		insert(k, info)
		findMin()
		deleteMin()
		decreaseKey(x, d)
		delete(x)
		totalLinks()
		totalCuts()
		meld(heap2)
		size()
		numTrees()

#### פלט

אין צורך בפלט למשתמש.

#### תיעוד

בנוסף לבדיקות אוטומטיות של הקוד שיוגש, קובץ המקור ייבדק גם באופן ידני. הקוד צריך להיות קריא, בפרט הקפידו על בחירת שמות משתנים ועל אורך השורות.

#### בדיקות

התרגילים ייבדקו באמצעות תוכנת טסטר שקוראת לפונקציות המפורטות מעלה בתרחישים שונים, ומוודאת את נכונות התוצאות. קובץ הטסטר שלנו לא יפורסם לפני הבדיקות. מומלץ מאוד לממש אוסף בדיקות עבור המימוש, לא בשביל ההגשה, אלא כדי לבדוק שהקוד לא רק רץ, אלא גם נכון! בקובץ שתגישו לא תהיה פונקציית **main** ולא יהיו הרצות קוד/הדפסות, דבר זה יפגע בטסטר שיבדוק לכם את התרגילים. אין צורך להגיש את הקוד הנוסף שכתבתם לחלק הניסויי.

## חלק ניסויי/תאורטי

- א. בתמונה המצורפת מופיעה למה מרכזית מהניתוח של ערימת פיבונאצ'י. נסחו מחדש את הלמה ביחס למימוש החדש של הערימה (תוך שימוש בפרמטר  $c$ ) והוכיחו אותה.
- ב. הוכיחו שהפתרון של נוסחת הנסיגה מהצורה  $a_n - a_{n-1} = a_{n-k}$  (עבור  $k$  קבוע) הינו  $a_n = \Omega(b^n)$  עבור  $b > 1$  כלשהו.
- ג. הסיקו כי הדרגה המקסימלית בערימה עם  $n$  איברים היא עדיין  $O(\log n)$  עבור כל  $c$  קבוע.

## Degrees in Maximally “damaged” trees

**Lemma 1:** Let  $x$  be a node of degree  $k$  and let  $y_1, y_2, \dots, y_k$  be the current children of  $x$ , in the order in which they were linked to  $x$ . Then, the degree of  $y_i$  is at least  $i-2$ .

בחלק זה נערוך שני ניסויים המשתמשים בערמת פיבונאצ'י. עבור כל ניסוי נמדוד את זמן הריצה, מספר החיבורים, מספר החיתוכים ומספר העצים בסיום. בכל הניסויים נגדיר  $n = 464646$ . לצורך שני הניסויים, בנו מערך מצביעים המחזיק באינדקס  $i$  מצביע לאיבר בעל מפתח  $i$ .

- ניסוי ראשון:
  1. נכניס לערמה ריקה את האיברים  $1, \dots, n$  בסדר אקראי.
  2. נמחק את המינימום.
  3. נמחק את המקסימום (בעזרת מצביע) עד שנישאר עם 46 איברים.
- ניסוי שני:
  1. נכניס לערמה ריקה את האיברים  $1, \dots, n$  בסדר אקראי.
  2. נמחק את המינימום.
  3. נבצע הקטנת מפתח של המקסימום (בעזרת מצביע) ל-0 עד שנישאר עם 46 איברים שהמפתח שלהם חיובי (וכל שאר האיברים הם עם מפתח 0).
  4. נבצע מחיקת מינימום שוב.

עבור כל ניסוי מבין השניים הנ"ל:

- הריצו אותו על ערכי  $c$  הבאים: 2,3,4,10,20,100,1000,5000.
- יש למלא בטבלה את הממוצע על פני 20 ניסויים זהים.
- בעזרת שימוש בערכים שמולאו בטבלה, יש להפיק כפלט גרף שמציג את זמן הריצה כפונקציה של  $c$ .

$c$	זמן ריצה (מילישניות)	גודל הערמה בסיום	מספר חיבורים	מספר חיתוכים	מספר עצים בסיום
2					
3					
...					

עבור כל אחד משני הניסויים, דונו בהשפעה של הערך  $c$  על התוצאות. דונו בהבדלים של התוצאות בין שני הניסויים ובהבדלים בין ההשפעות של הערך של  $c$  על התוצאות בניסוי הראשון לבין ההשפעות של הערך  $c$  על התוצאות בניסוי השני.

**הוראות ההגשה מופיעות בעמוד הבא.**

## הוראות הגשה

הגשת התרגיל תתבצע באופן אלקטרוני באתר הקורס במודל.

הגשת התרגיל היא בזוגות בלבד!

כל זוג יבחר נציג/ה ויעלה רק תחת שם המשתמש של הנציג/ה את קבצי התרגיל (תחת קובץ **zip**) למודל.

על ההגשה לכלול שלושה קבצים:

1. קובץ המקור (הרחבה של קובץ השלד שניתן) תחת השם **FibonacciHeap.java**.
2. קובץ טקסט **info.txt** המכיל את פרטי הזוג באנגלית: מספר ת"ז, שמות, ושמות משתמש.  
**אם מגישים בשלושה אז הסטודנט שצריך להופיע ראשון במסמך הוא הסטודנט מבין השלושה שזכאי להגיש בשלושה (מילואימניק או מישוהו שקיבל אישור פרטני להגיש בשלושה בגלל סיבה אחרת).**
3. מסמך תיעוד חיצוני, המכיל גם את תוצאות החלק הניסויי. את המסמך יש להגיש בפורמט **pdf**.

שמות קובץ התיעוד וקובץ הzip צריכים לכלול את שמות המשתמש האוניברסיטאיים של הזוג המגיש לפי הפורמט **FibonacciHeap\_username1\_username2.pdf/zip**, בתוכן הקבצים יש לציין את שמות המשתמש, תעודות הזהות ושמות המגישים (בכותרת המסמך ובשורת הערה בקובץ המקור).

הגשת שיעורי הבית באיחור - באישור מראש בלבד. הגשה באיחור ללא אישור תגרור הורדת נקודות מהציון. הגשת התרגיל היא חובה לשם קבלת ציון בקורס.

בהצלחה!